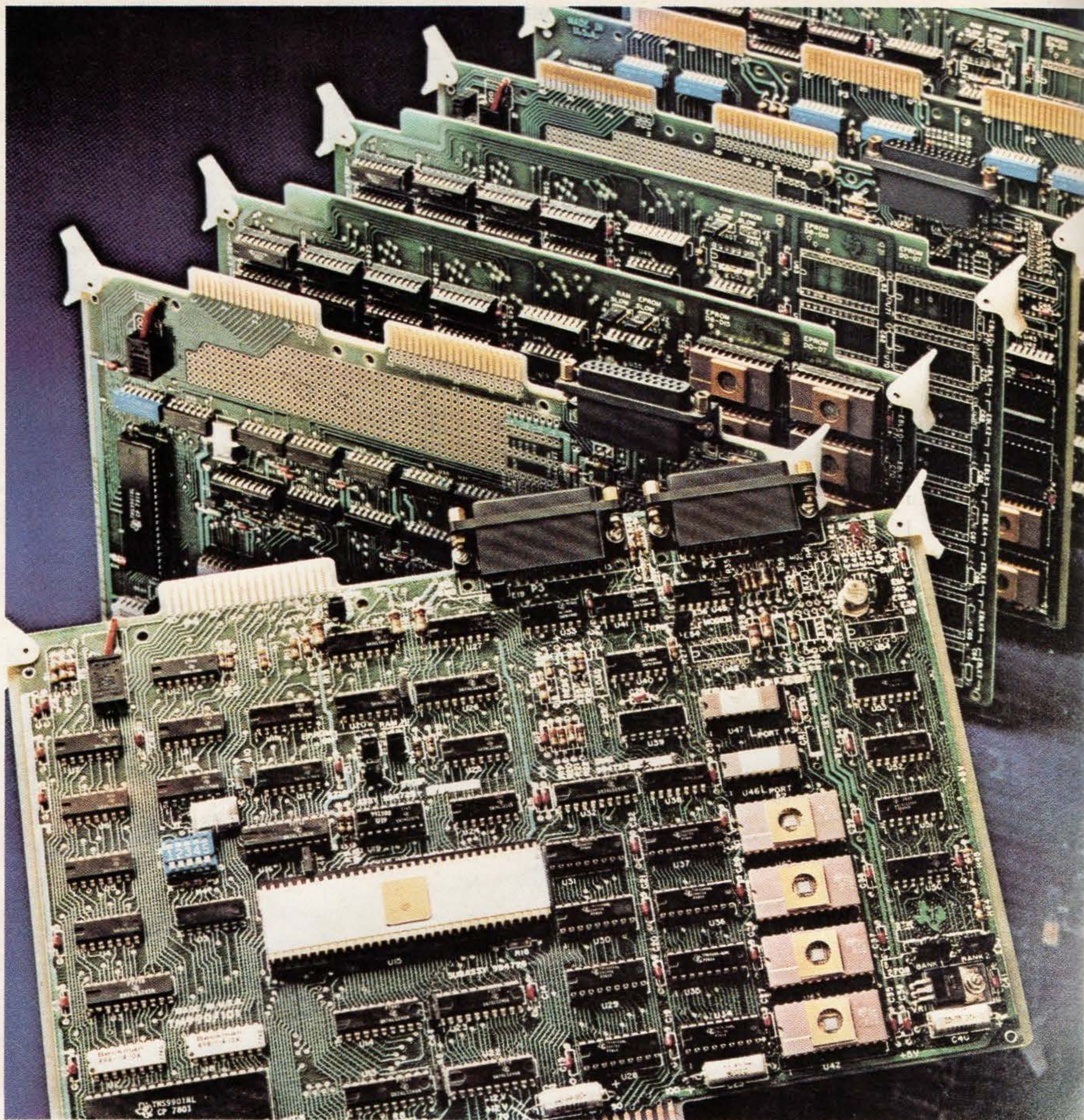
UNA PUBBLICAZIONE
DEL GRUPPO EDITORIALE JACKSON

ANNO 2 N. 3

Bit

MICROPROCESSORS-HARDWARE - SOFTWARE
HOME & PERSONAL COMPUTERS

L. 2000

COMPUTER

La Texas Instruments annuncia il primo modulo di microcomputer a 16 BIT con linguaggio BASIC. Più facile da programmare, ampliato nella capacità di memorizzazione e di comunicazione

Sono disponibili presso i distributori autorizzati T.I., dei veri e propri computers su singola scheda.
Stiamo parlando dei nuovi moduli di microcomputers TM 990/101M.
La maggiore estensione di memoria, la possibilità di comunicazioni simultanee su due porte (con protocolli di comunicazione già implementati su board), e la disponibilità del POWER BASIC, linguaggio ad alto livello che rende estremamente veloce la programmazione, sono la caratteristiche fondamentali di questi moduli.

Questi nuovi moduli della Texas Instruments riducono i tempi di progetto e di sviluppo, il numero di

componenti del sistema ed i costi di realizzazione.

Queste unità sono preassemblate, pretestate e pronte all'uso, e questo garantisce un'alta *affidabilità al sistema*.

Estensione di memoria quattro volte più grande

I microcomputers TM 990/101M possiedono una capacità di memoria RAM statica su board che va da 1K a 2K parole da 16 bits.
L'estensione di EPROM va da 2K a 4K parole da 16 bits.

Capacità di comunicazione doppia

Sul board sono disponibili due porte per comunicazioni seriali: una usata per utilizzazioni "remote" (terminali o modem), l'altra per utilizzazioni "locali" (terminali EIA, teletype, microterminale TM 990/301 della T.I.).

Il meglio nell'ambito dei microcomputers

Nella serie TM 990 della T.I. troverete la più ampia scelta possibile per prestazioni/costo, di microcomputers a 16 bits che possono soddisfare le vostre necessità a livello di sistema.
Ideali per valutazioni su microprocessori e per accelerare il progetto di sistemi a microprocessori, rappresentano una valida alternativa in produzione.
Caratteristica fondamentale di tutti i moduli è la completa compatibilità a livello di set di istruzioni con gli altri membri dell'intera famiglia TM 990/9900.

Per valutazioni ed applicazioni OEM

- TM 990/100M - Utilizza il microprocessore TMS 9900 a 16 bit in tecnologia N MOS, della Texas Instruments.

  1K bytes di RAM statica, 2K bytes di EPROM, e porte di I/O seriali e parallele programmabili, fanno di questa unità un potente microcomputer su singola scheda.

- TM 990/180M - Utilizza il microprocessore TMS 9980 a 16 bit, in tecnologia N MOS, della T.I., che consente di lavorare a 2,5 MHz, e che contengono una interfaccia a 8 bit per la memoria, minimizza i costi dei sistemi più piccoli.

Per espansioni di memoria

- TM 990/201 - 8K bytes di EPROM e 4K bytes di RAM statica. Espandibile a 32K bytes di EPROM e 16K bytes di RAM.
- TM 990/206 - 8K bytes di RAM statica espandibile a 16K bytes.

Per ingresso dati e monitoraggio

- TM 990/301 - Consente l'ingresso di dati di programma, la visualizzazione e la modifica del contenuto dei registri interni e della memoria sotto controllo software (TIBUG).

Per espandere le linee di I/O

- TM 990/310 - Modulo di espansione delle linee di I/O a 48 bit.

Per sviluppare software e per la produzione

- TM 990/401 - Monitor interattivo di debug (TIBUG) già programmato in EPROM.
- TM 990/402 - Assemblatore line-by-line già programmato in EPROM.
- TM 990/450 - POWER BASIC già programmato in 8K bytes di E-PROM.
- TM 990/451 - POWER BASIC già programmato in 12K bytes di E-PROM.

Sono inoltre disponibili una vasta gamma di connettori e cavi per gli usi più comuni, schede di prototipizzazione, ed uno chassis a 4-slot (TM 990/510) già predisposto per ospitare 4 schede della famiglia TM 990.

Supporto software

I moduli di microcomputer della serie TM 990 sono completamente supportati dal sistema di sviluppo AMPL (Advanced Microprocessor Prototyping Laboratory) della Texas Instruments.

L'AMPL consente di effettuare trace a 10 MHz ed emulazioni per i microprocessori TMS 9900, 9980, SBP 9900, TMS 9940 e per tutti i *microprocessori della T.I. che saranno* prodotti in futuro.

L'AMPL è disponibile nella configurazione con sistema a floppy disk o in quella con sistema a dischi rigidi in modo da soddisfare le più svariate esigenze degli utenti. I programmi possono essere editati, assemblati, lincati, caricati ed eseguiti più velocemente che con i sistemi tradizionali a nastri o a cassette.

L'emulazione del TMS 9900/9980 consente di sviluppare e di debuggare il software direttamente su un modulo TM 990 visualizzando e controllando, contemporaneamente, le operazioni per mezzo del sistema di sviluppo AMPL.

Per salvaguardare il progetto di oggi dalle innovazioni di domani

I microcomputers della serie TM 990 e l'AMPL si integrano perfettamente con l'intera famiglia 990/9900 rendendola veramente completa e di estema potenzialità nell'ambito delle applicazioni a 16 bit.

Questa famiglia è composta da una vasta gamma di microprocessori, microcomputers e minicomputers che usano tutti la stessa architettura orientata verso la memoria, lo stesso set di istruzioni; sono tutti completamente compatibili e sono previsti per essere supportati dallo stesso sistema di sviluppo.

Con la famiglia 990/9900 il progettista può scegliere la soluzione a costo più basso, e più soddisfacente alle sue esigenze, senza preoccuparsi degli sviluppi futuri del suo prodotto, in quanto la completa compatibilità software rende il progetto software direttamente utilizzabile su prodotti che si possono diversificare per sofisticazione hardware.
Per maggiori informazioni contattate i nostri distributori autorizzati oppure l'ufficio Promozione Commerciale, Cittaducale - RIETI

REDIST
presenta:

CENTRONICS
Microprinter-P1

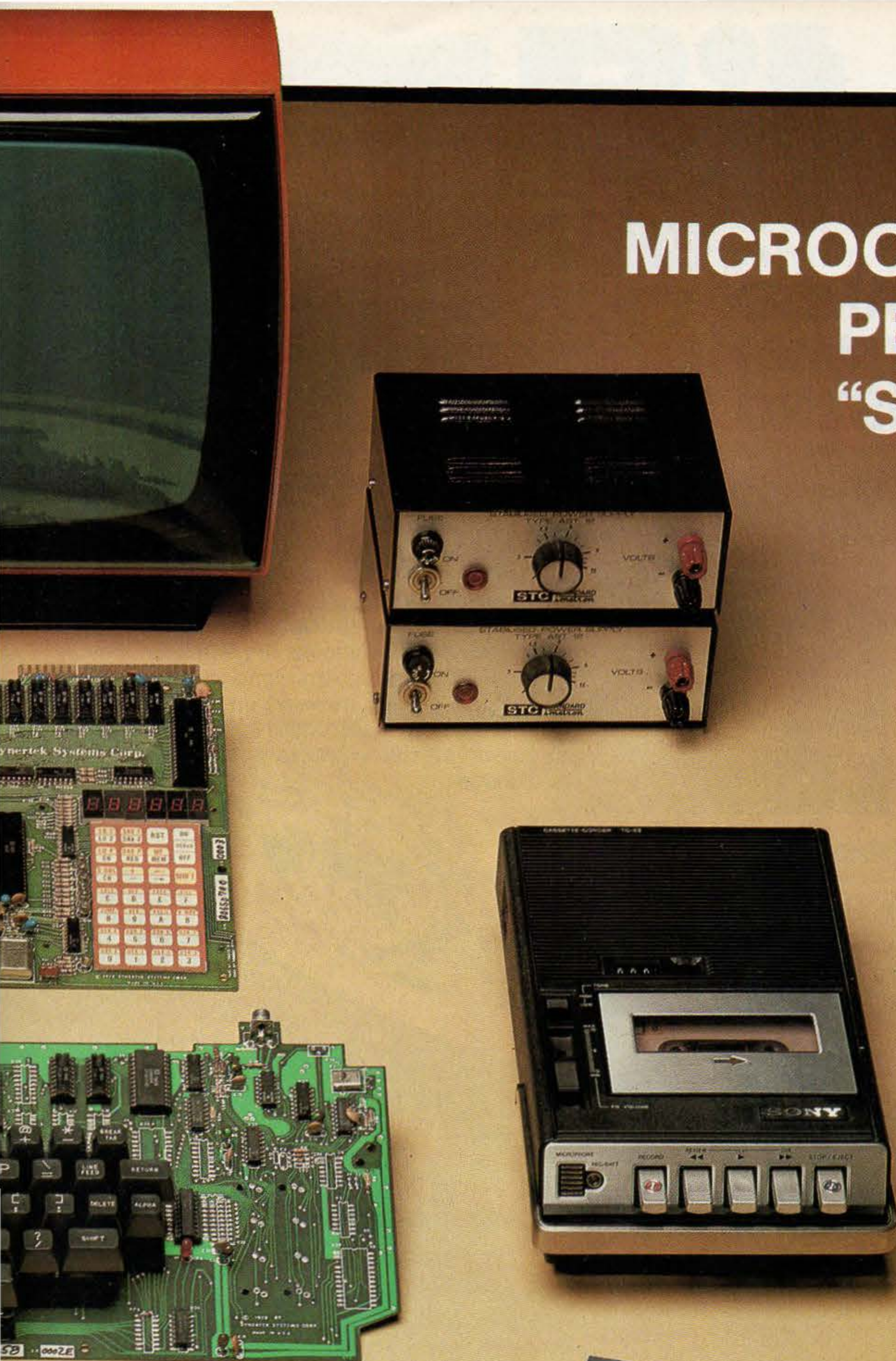
Synertek Systems Corp.

HARDWARE

□ Sistema basato sulla CPU 6500, il microprocessore più venduto nel mondo.
□ Facilità d'uso
L'interfaccia verso l'utente è costituita da una tastiera ASCII completa e da un TV domestico. Inoltre: Set di 128 caratteri grafici, possibilità di video inverso, controlli completi di cursore, velocità fino a 9600 baud.
□ Archivio programmi
I programmi possono essere memorizzati su cassette maghetiche audio (velocità di registrazione 1200 baud) e successivamente richiamati nel sistema al momento della esecuzione.
□ Emissione documenti
Programmi e risultati dell'esecuzione possono essere trasferiti ad una stampante seriale veloce.
□ Espandibilità del sistema
Fino a 4K RAM, 16K ROM, 4 porte di I/0 per pilotare carichi di potenza.
□ Alimentazione
L'intero sistema viene alimentato da una sola tensione (+ 5V)
□ Il sistema è quindi in grado di fornire prestazioni molto sofisticate a basso costo.

SOFTWARE

□ Monitor/Debugger (4K ROM)
Possibilità di single-step, breakpoint a software, protezione in scrittura della memoria, ricerca byte, espandibilità con programmi di utente, ecc.
□ Basic (8K ROM)
Basic Esteso, Floating-point a 9 digit, Funzioni scientifiche, Funzioni di formattazione, access in lettura/scrittura a qualsiasi locazione di memoria, possibilità richiamare sub-routines Assemble
□ Assembler/Editor (8K ROM)
Assembler: Possibilità di macro, ingresso da RAM o audio-cassett produce oggeti rilocatabili, Loade

MICROCOMPUTER PERSONALE "SYNERTEK"

rilocatabili, assemblaggi con listing dell'intero sorgente o dei soli errori.
Editor: ricerca stringhe, inserzioni blocchi, inserzione e cancellazione di linea, tabulazione automatica in printout, possibilità di hard-copy, ecc.

MANUALI

2 Manuali in Italiano - "Guida alla programmazione" - "manuale d'uso".
5 Manuali in Inglese
- "SYM Reference Manual"
- "KTM 2 Reference Manual" "Basic Reference Manual"
- "Programming Manual"
- "Editor/Assembler Reference Manual"

Distribuiti da REDIST
divisione della G.B.C. italiana
Viale Matteotti, 66 - 20092 Cinisello Balsamo
Tel: 02/6189391 - 6181801 - Telex: 36028 GBC MIL

SOMMARIO

In copertina:
Il Personal Computer è per tutti.
Grafica di Renato NISI.

INDICE INSERZIONISTI

DIRETTORE RESPONSABILE
Giampietro Zanga

COORDINATORE TECNICO
Marcello Montedoro

CAPO REDATTORE
Dino Bortolossi

SEGRETARIA DI REDAZIONE
Cecilia De Serio

GRAFICA E IMPAGINAZIONE
Job Line srl

CONSULENZA
E COORDINAMENTO
Marcello Montedoro

DIFFUSIONE E ABBONAMENTI
Gabriella Napoli, Silvia Decari

DIREZIONE, REDAZIONE
P.le Massari, 22 - 20125 Milano
Telefoni 68.03.68 - 68.00.54

AMMINISTRAZIONE
Via Vincenzo Monti, 15 - 20123 Milano

PUBBLICITA': Concessionario per l'Italia e l'Estero Reina & C. S.n.c. - Via Ricasoli, 2 - 20121 Milano Tel. (02) 803101-866192 Via Saverio Carmignano, 10 - 00151 Roma Tel. (06) 5310351

FOTOCOMPOSIZIONE: New Comp Via S. Michele al Carso, 5 - Nova Milanese

STAMPA: Litografia del Sole srl Buccinasco

Concessionario esclusivo per la diffusione in Italia e all'Estero: SODIP - Via Zuretti, 25 - 20125 Milano Via Serpieri, 11/5 - 00197 Roma

Spedizione in abbonamento Postale Gruppo IV/70

Prezzo della rivista L. 2.000

Numero arretrato L. 3.000
Abbonamento annuo L. 8.000
per l'Estero L. 12.000

I versamenti vanno indirizzati a: Jackson Italiana Editrice S.r.l. Via V. Monti, 15 - 20123 Milano mediante emissione di assegno bancario, cartolina vaglia o utilizzando il c/c Postale numero 11666203 Per i cambi d'indirizzo, indicare, oltre naturalmente al nuovo, anche l'indirizzo precedente, ed allegare alla comunicazione l'importo di L. 500, anche in francobolli.

EDITORIALE

Appuntamento a "Bit '79"

Una volta l'accesso al computer era poco meno che uno status symbol: dopo un certo periodo di tirapiedaggio, uno finiva con l'essere considerato degno di accedere alla Grande Macchina Che Pensa, e pagava quest'altro onore con l'essere trattato da seccatore ad ogni minima, timida richiesta rivolta ai Bianchi Sacerdoti del Computer. Si rifaceva poi a casa, asfissiando i poveretti che avevano la sfortuna di essergli parenti con interminabili esagerazioni sulla suprema velocità, intelligenza e potenza del Grande Mostro, oltre che naturalmente sulla sua modesta abilità personale nel far fronte a situazioni cibernetiche quasi prive di via d'uscita. Dopo un po' l'unico commensale rimasto, la povera nonnetta sorda come una campana, come vedeva le labbra del narratore fermarsi chiedeva un po' trasognata - "E i genitori di lei, che ne pensano? Non siete un po' troppo giovani?" -.

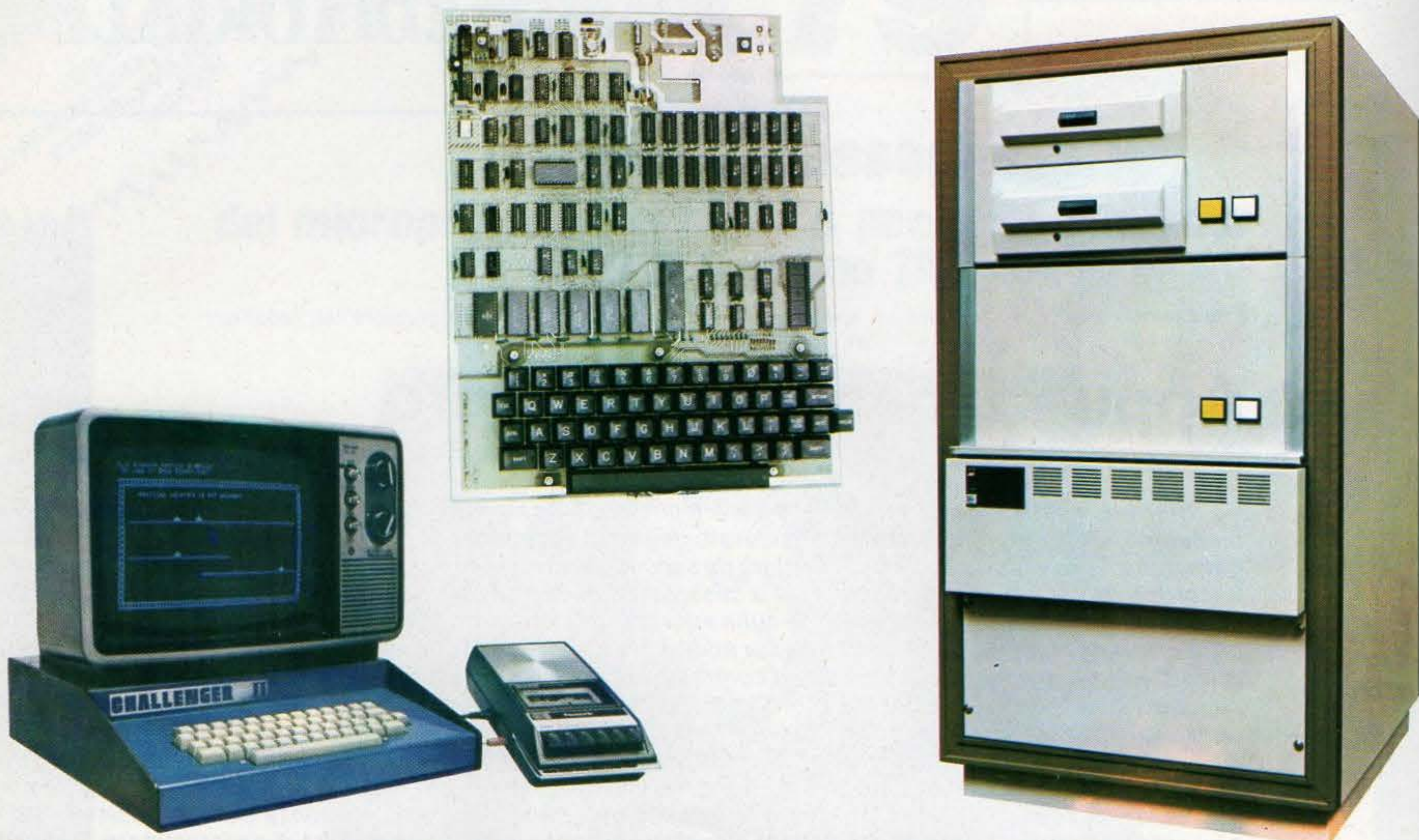
Erano tempi duri. Non appena uno si accorgeva di aver preso gusto a programmare, si ritrovava legato mani e piedi al datore di lavoro il quale, intuito il suo punto debole, per fargli fare le cose più assurde lo minacciava di assegnarlo ad un altro lavoro più prestigioso, megli retribuito, ma senza Computer. Drammatico. Licenziarsi era più difficile di smettere di fumare, perché significava smettere di programmare. E anche mentre uno era immerso nei suoi profondi pensieri programmativi, a un pelo dall'orgasmo logico circonvoluzionale, un pensiero si insinuava fastidioso in background: tutto questo lo posso fare io, dimostrando a me e al mondo la mia potenza psichica, ma, ahimé, non è mio e mi può esser tolto in qualsiasi momento. C'è ancora chi sostiene che la programmazione non è una droga, ma si tratta di una minoranza che ancora non ha imparato a programmare.

Per fortuna usi e situazioni si evolvono. Prima il timido pensiero che, in fondo, un computer non è che una certa quantità di hardware messa insieme con abilità, e che l'hardware non costa poi moltissimo. Poi la clamorosa sconfitta delle costosissime memorie a nuclei di ferrite da parte delle RAM e delle EPROM (molti hanno versato lacrime di sollievo sulla prima, rudimentale 1103, 1 K bits SENZA NUCLEI!). Poi ancora i microprocessors, dapprima limitati e spartani nel firmware, ma autonomi. Da qui agli home computers il passo è stato breve, giusto un battito di palpebre ed un paio di tentazioni e pensieri eretici: non più code al centro di calcolo - suggeriva un piccolo demonio interno - né lunghe tirate e lacrime al direttore perché servono 4 K in più, né un rigido orario di lavoro per cui abbandonare il problema sul più bello, quando si intravvede la soluzione. Aveva ragione, quel diavolo di demonio interno.

Naturalmente con i computers si sono ristrette anche le periferiche: non a senso una CPU da centomila lire con un disco da quindici milioni. Ed ecco su un altro fronte unità periferiche più semplici e meno costose come floppy disk, minicassette, CRT controllers e displays appena un po' fuori della portata di tutte le tasche, giusto quel tanto da invogliare un poveraccio a privare se stesso e la famiglia del superfluo per pagarsi la droga.

Oggi home e personal computers stanno esplodendo, tutti li cercano, tutti li vogliono. Noi alla Jackson ovviamente lo sapevamo e sulle prime abbiamo cercato di far loro un po' di spazio su Elettronica Oggi. Ma è stato come quando si invita a casa un amico e questi capita con la moglie e i sette figli: non sapevamo dove metterli tutti. Abbiamo dovuto dapprima creare un'altra rivista, questa che state leggendo, tutta per loro. Ma non è bastato. Un computer del costo complessivo di una trentina di milioni lo si può anche solo guardare sulle pagine di una rivista, home e personal computers sono troppo vicini alla portata delle nostre tasche per accontentarsi. Uno desidera vederli, toccarli e quando nessuno guarda, premere un pulsante per vedere cosa succede (poi magari succede solo che si spengono le luci della toilettes, ma non importa). Beh, eccovi accontentati. Abbiamo organizzato per voi una mostra dei personal e home a Milano dal 6 al 9 giugno pressi i locali del Centro Commerciale Americano - Via Gattamelata, 5, ci vedremo là. Almeno noi vi vedremo, voi probabilmente sarete troppo presi dalla muta contemplazione degli oggetti dei vostri desideri per vedere alcunché d'altro.

Gianni Dangru

State pensando ad un microcomputer?

Assicuratevi di esaminare le offerte di prodotti della più grande Ditta del mondo, produttrice di una linea completa di Microelaboratori. Tutte le macchine della Ohio Scientific sono dotate del più veloce e completo Basic-In-Rom o Su-Disco, ad impiego istantaneo, esistente nel campo della microelaborazione.

Serie Challenger I

Sistemi di elaborazione economici che parlano in Basic. Ideale per studenti computer amatori, la scuola, la casa.
Superboard II - Il primo sistema al mondo completo, su una scheda comprendente tastiera, interfaccia per video display e audio-cassetta, BASIC-IN-ROM e fino a 8 K di memoria RAM. Challenger IP - Superboard II completo di contenitore e con alimentatore incorporato. Challenger IP - Disk - Sistema completo di mini-dischetti espandibile fino a 32 K di memoria RAM.

Serie Challenger II P

Microelaboratori, BUS orientati, ad altissime prestazioni, per impieghi personali, scolastici, di ricerca e nelle piccole aziende.
C2 - 4 P - Il portatile professionale. C2 - 8 P - La macchina personale più espandibile del mondo per impieghi commerciali e di ricerca.
C2 - 4 P - A mini - dischetti il meglio fra i portatili. C2 - 8 P - A due dischetti per piccola azienda a un prezzo imbattibile.

Serie Challenger II con interfaccia seriale

Ha le stesse grandi caratteristiche della serie Challenger II P per coloro che hanno terminali seriali, utilizzata da piccole aziende, scuole, industrie.

Challenger III il meglio e il più potente nel campo dei microelaboratori

L'unico sistema a 3 processori (6502, 6800, Z80) per soddisfare le più sofisticate esigenze delle imprese. C3 - OEM - Il sistema a due dischetti compatto per grandi volumi di lavoro con 32 K di memoria RAM. C3 - B - Il sistema basato sul disco Winchester da 74 M bytes. Il microelaboratore più potente del mondo. La Ohio Scientific vi offre anche la più ampia gamma di accessori per l'espansione e la più grande scelta di software economico. Confrontate un modello della Ohio Scientific con un modello comparabile di un'altra Casa che avete preso in considerazione. Confrontate le prestazioni, la reale possibilità di espansione, il software e i prezzi, e capirete perchè siamo diventati la più grande società di microelaboratori che produce la linea di sistemi più estesa del mondo.

PRODUTTORE

OHIO SCIENTIFIC

DISTRIBUTORE INTERNAZIONALE

AMERICAN DATA-HOME AND OFFICE COMPUTER

EDICONSULT

CONCESSIONARIE PER L'ITALIA:

SOCORED s.a.s. - Via Mandelli, 2 - CARUGATE - Tel. 02-294081-298081

PIEMONTE SED - Via Umberto I, 32 - S. Daniele (UD) Tel. 0432 - 482000

I.C.S. INFORMATICA COMPUTER SERVICE di Ciccolini s.a.s. - V.le C. Battisti, 13 - 62024 MATELICA (MC) Telef. 0737 - 8714-800914

EDILCONSULT - Via Andrea Busiri Vici 5 - ROMA - Telef. (06) 55.77.116

Distributore esclusivo per l'Italia:

EDICONSULT s.r.l.

Via Caccini, 12 - 20052 MONZA - Tel. 039/389850

Nome ....

Indirizzo ....

Tel. ....

Città .... **CAP** ....

Per ulteriori informazioni allegare la somma di Lire 500.

Bit

Prossimo futuro

L'utilizzo di sistemi computerizzati è un'esigenza ormai avvertibile in modo diffuso in vari settori; d'altra parte la drastica riduzione del prezzo dei sistemi porta a nuovi campi d'applicazione.

Il fenomeno del personal computer illustra in maniera esemplare questa nuova realtà: le aree in cui il personal computer è o può essere operante in termini immediati sono note, l'allargamento a nuove utenze o a nuovi servizi è una progressione abbastanza naturale.

Da qui a fare previsioni il passo è breve, ma con dei rischi: il primo è quello di vedersi superati da una realtà che ha i tempi di sviluppo estremamente rapidi, l'altro è quello di sottovalutare problemi concettuali e realizzativi che ogni campo di applicazione, specialmente se nuovo, porta con sè, trascinati dagli sviluppi incalzanti delle innovazioni tecnologiche.

In situazioni analoghe ci si è trovati nel passato quando si sono fatte valutazioni poi dimostratesi errate nel considerare le possibili applicazioni di determinate tecnologie; probabilmente anche noi corriamo questo rischio.

Con questa premessa è necessario prendere nota di indicazioni estremamente significative che stanno emergendo nell'area del personal computer (chiaramente negli U.S.A.) e che vedono in prospettiva l'utilizzo di tali sistemi come terminali intelligenti con cui l'utente generico può accedere, tramite linea telefonica o onde radio (soluzioni meno flessibile della precedente), a banche di dati diversificate sia come natura delle informazioni in esse contenute, sia come enti che creano e gestiscono tali sorgenti di informazioni.

Le problematiche connesse a questo approccio sono molto complesse e di varia natura.
Ad ogni buon conto veniamo ai fatti. La Apple Computer Inc., che produce un personal computer, l'APPLE II, molto diffuso in America, ha annunciato recentemente un'intesa con la Dow-Jones Company Inc., in base alla quale la Dow-Jones permetterà l'accesso ai propri archivi (quotazioni di borsa e informazioni similari) agli utenti Apple Computer supportati a tal fine da programmi forniti dalla Apple Computer stessa: infatti questa società sta preparando una serie di programmi per l'uso del servizio, il primo dei quali permetterà all'utente di ottenere le quotazioni di tutte le azioni a listino nelle sei maggiori borse americane. Il servizio prevede una tariffa iniziale di contratto, quindi una tariffa legata al tempo di utilizzo del servizio stesso.

Questa iniziativa, come altre similari, fa parte della Personal Computer Network (PCNET), che ha un organo di supervisione (Commissione PCNET), che ha fissato i protocolli per le comunicazioni tra personal computer che utilizzano la rete telefonica, ed altre regolamentazioni necessarie.

Altre strutture ed altro servizio sono forniti da un'iniziativa realizzata nell'area di Chicago da hobbisti di personal computers. A questo servizio può accedere chiunque abbia un terminale personal computer con accoppiatore acustico verso la linea telefonica. Il sistema di gestione del servizio comprende le funzioni di ricezione messaggi, accesso al sommario per contenuti dei messaggi e quindi accesso al messaggio stesso: messaggi tipici sono annunci di vendita/scambio di hardware e software, informazioni relative a clubs di computers, inserzioni di lavoro, ricerca di persone con interessi simili, etc.

Le indicazioni che risultano da questi brevi cenni sono molto chiare: il personal computer non è un giocattolo sofisticato, ma il mezzo con cui nel prossimo futuro le esigenze di comunicazioni delle informazioni potranno trovare un'adeguata risposta.

GIM: aumenta l'assemblaggio di memorie in Europa

Con il ristagno sul mercato consumer, la General Intrument Microelectronics è stata costretta a correre a correre ai ripari, studiando nuovi prodotti ed individuando nuovi sbocchi. Fra gli orientamenti che vanno prevalendo si possono già ora indicare come certi quelli che riguardano le applicazioni interessanti i seguenti settori: elettrodomestico (apparecchi bianchi), controllo riscaldamento industriale, il giocattolo (inteso in senso ampio non essenzialmente videogame).
Quanto ai prodotti si può con certezza affermare che la GIM darà impulso all'attività inerente alle memorie, accentuando la produzione svolta in Europa. Alle ROM da 16 e 32K, oggi costruite anche nel nostro continente, la società programma, ad esempio, di aggiungere le EAROM. Di queste memorie la GIM avrebbe venduto circa 2,5 milioni di prezzi, tutte da 1400 bit se destinate all'industria televisiva. Ora progetta con le stesse EAROM di entrare nell'area dell'informatica, da qui lo sviluppo di unità con una capacità di 3400 bit. La fabbricazione troverà accoglienza a Glenrothes (Scozia) in un impianto di cui è in corso la ultimazione. La GIM sta inoltre valutando le opportunità di assemblare in Europa, allegerendo la dipendenza dell'oriente.

Mostek: investimenti per 80 mdi $ in Irlanda

Dopo le voci finalmente la Mostek ha ufficilizzato la decisione di investire 80 milioni di $ nella costruzione di una fabbrica di semiconduttori in Irlanda. Lo stabilimento di Blanchardstown, nelle vicinanze di Dublino, rappresenta la prima iniziativa produttiva della Mostek nel mercato comune europeo. In una conferenza stampa con i dirigenti dell'Industrial Development Authority (IDA) of the Republic of Ireland, esponenti della Mostek hanno effermato che nel nuovo complesso irlandese verranno prodotti circuiti integrati di memoria, microprocessori e circuiti per applicazioni industriali.
Il primo passo del programma di produzione, articolato in tre fasi, avrà inizio queste primavera in uno stabilimento esistente situato a Cherry Orchard, mentre si procederà alla costruzione di una fabbrica di quasi 10.000 MQ. A Blanchardstown, dove la Mostek ha acquistato un terreno di 20 ettari.
La prima fase del progetto Mostek in Irlanda comporterà il test finale e la classificazione di complessi circuiti integrati. La Mostek passerà alla seconda fase, che includerà l'assemblaggio automatico dei circuiti integrati, una volta completata la costruzione della fabbrica di Blanchardstown.
L'ultima fase del progetto è la produzione di wafer di silicio, che richiederà la costruzione di altri 18.000 MQ. Sempre a Blanchardstown entro il 1982.
La Mostek, fondata nel 1969 a Carrollton, vicino a Dallas, nel Texas, dà oggi lavoro ad oltre 5.000 dipendenti. Le vendite nel 1978 hanno superato i 130 milioni di dollari con un incremento del 230% rispetto al 1976. Oltre alla casa madre a Carrollton, la Mostek possiede uno stabilimento di assemblaggio in Malesia. Il nome della Mostek deriva dalla tecnologia MOS (Metal-Oxide-Semiconductor), su cui sono basati molti dei suoi prodotti. Le applicazioni dei microprocessori Mostek sono in continuo aumento, inoltre la società ha introdotto una serie di componenti di memoria ad alta densità e con prestazioni migliorate.
Le memorie RAM da 4K e da 16K della Mostek per la filosofia di progetto e le prestazioni elevate, sono diventate da tempo standard industriali.

La SGS-ATES si estende alla microinformatica

La SGS-ATES modifica la propria posizione strategica nel campo dei microprocessori: da quella di secondo fornitore a quella di produttore di microsistemi completi. Tale cambiamento è motivato dalla volontà di andare incontro alla domanda crescente di servizi di supporto a tutti i livelli. E' per questo che, dopo una accurata valutazione di tutti i sistemi a microprocessore disponibili e delle necessità di una loro utilizzazione non limitata al presente, è stata presa la decisione di allargare la portata dell'accordo con la Zilog Corporation of America.

Il nuovo contratto (di cui non sono stati resi noti gli aspetti finanziari) comprende anche i diritti per la distribuzione dei sistemi di sviluppo per Z-80 -Z8 e Z8000 (lo ZDS 1/25 e 1/40).

La SGS-ATES renderà disponibile l'intera documentazione tecnica sullo Z80, nelle lingue europee più diffuse, e organizzerà una serie di corsi in tutta Europa. Grazie a tali cambiamenti ed alla possibilità di fornire d'ora in poi tutti i servizi di assistenza e supporto, la SGS-ATES offre all'Europa quei medesimi servizi, di cui soltanto l'utilizzatore americano di sistemi basati sullo Z80 poteva prima usufruire. I motivi che hanno spinto la SGS-ATES ad estendere il proprio impegno con la Zilog, vanno al di là del successo ottenuto come sorgente alternativa dello Z80.

Al fine di assicurare peraltro una presenza continua sul mercato, il nuovo contratto stipulato tra SGS-ATES e Zilog prevede, per la società italiana, i diritti di unico produttore europeo dei microprocessori Z8000 e Z8.

Questo accordo avrà certamente una grande importanza sia per la Zilog sia per la SGS-ATES, dando all'una l'appoggio della solida e provata esperienza di un costruttore europeo, e all'altra un ideale trampolino di lancio in un mercato competitivo e in forte espansione. C'è un aspetto che di questo nuovo contratto va sottolineato: diversamente da quanto era accaduto per lo Z80 e per altri μP costruiti dala società di Agrate come seconda sorgente, questa volta SGS-ATES e Zilog partono insieme. I vantaggi dovuti allo sviluppo del sistema da parte della società americana sono compensati dalla maggiore introduzione e dalla esperienza di produzione della casa italiana. C'è solo da augurarsi che le difficoltà createsi ai vertici della Zilog non provochino contraccolpi sui programmi di crescita.

Secondo store della DEC

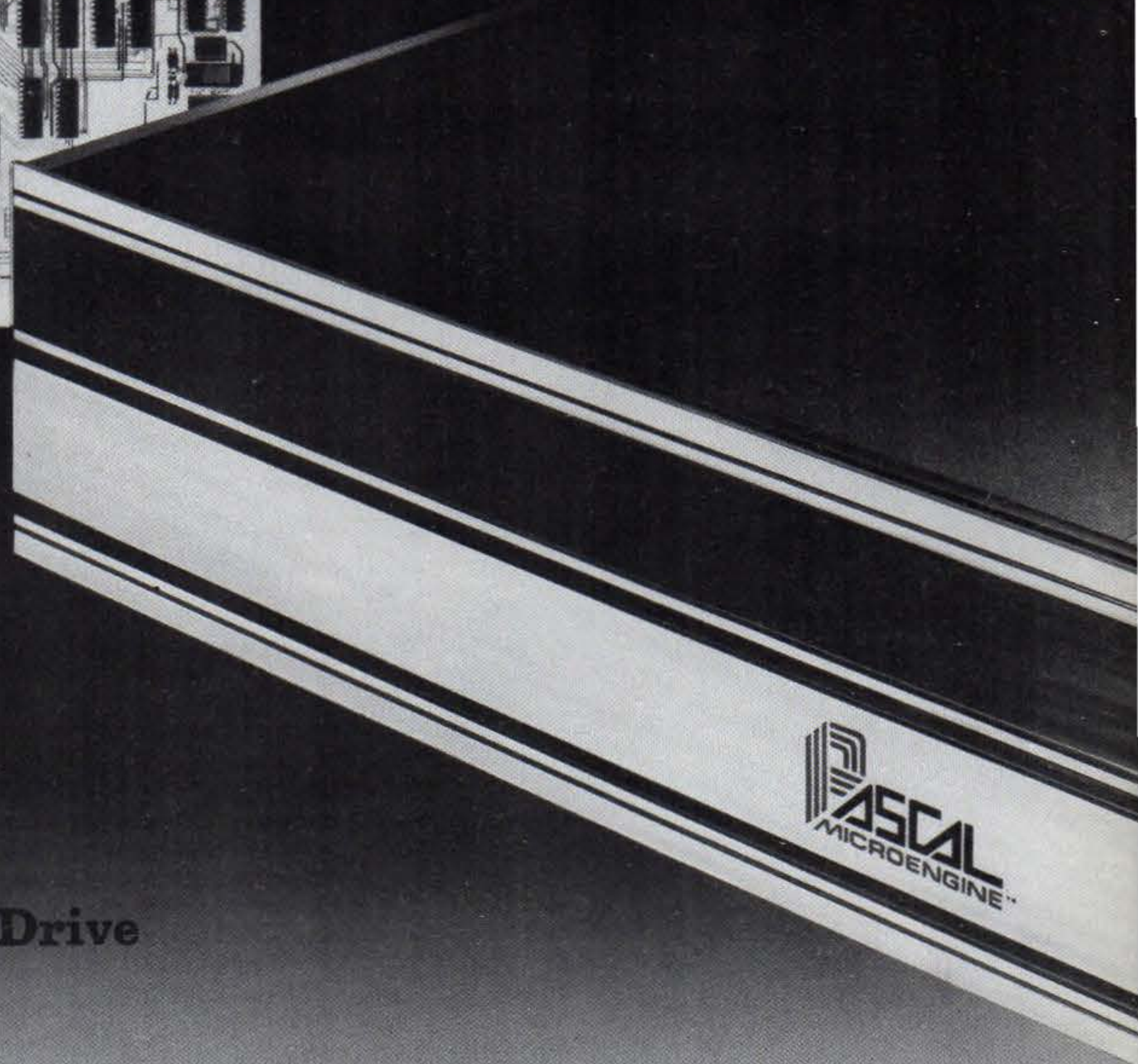
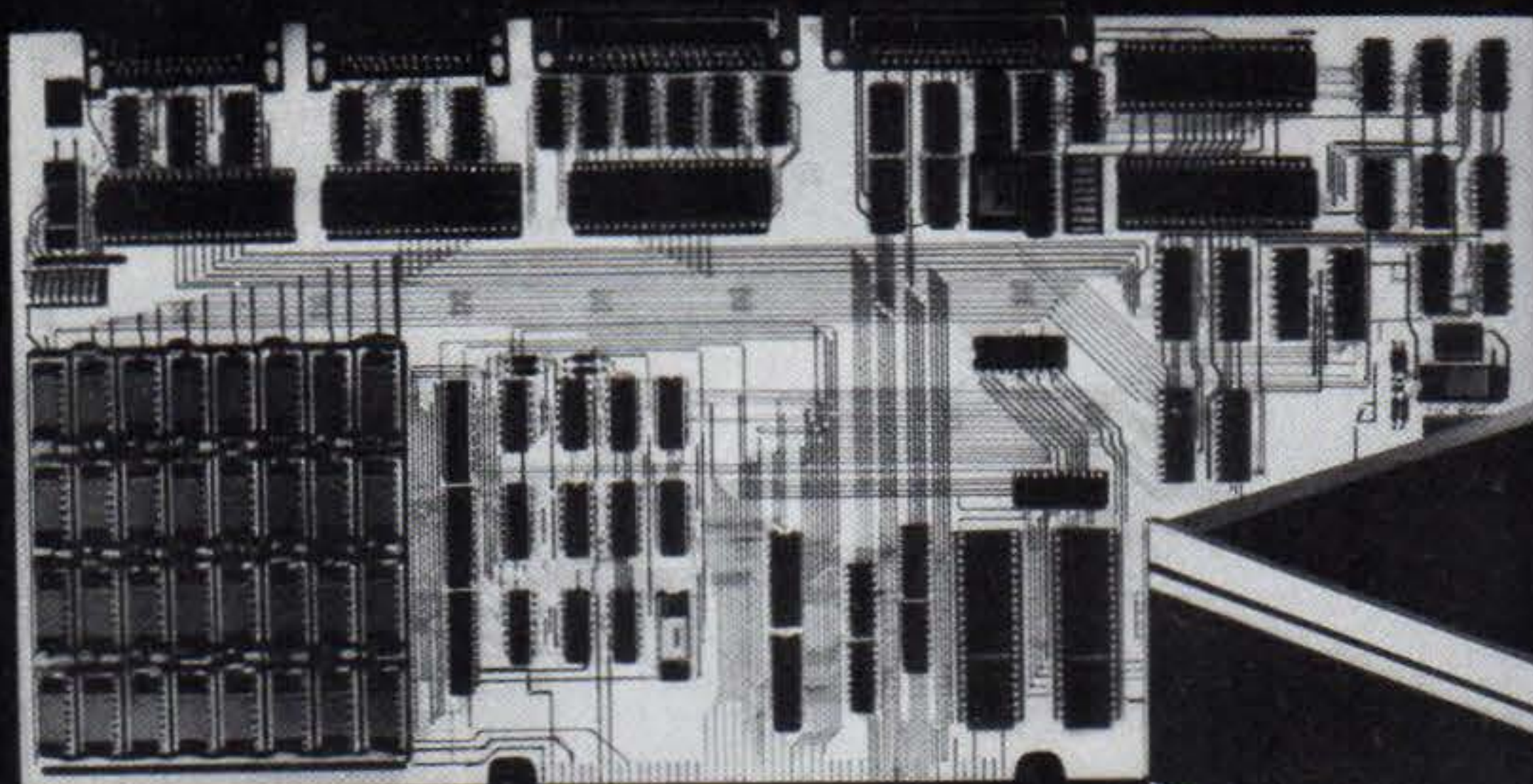
Nello scorso marzo la DEC ha aperto un secondo punto di vendita al dettaglio nella città di Boston per vendere sistemi di elaborazione e di word processing per piccole applicazioni gestionali (il primo era stato aperto a Manchester - New Hampshire nel 1978). Il presidente K. H. Olsen ha riferito in sede di commento ai risultati conseguiti nei primi nove mesi (vendite per 1279 milioni di $ e utili per 116,7 milioni), che i programmi Digital prevedono l'apertura di numerosi punti di vendita di questo tipo nelle più importanti aree metropolitane entro l'anno.

L'APL contro Basic

Tra i fabbricanti di personal computer una corrente di interesse va sviluppandosi in favore del linguaggio APL che parecchi considerano, oggi, più promettente del Basic. Il primo fabbricante ad optare e fra i più convinti sostenitori dell'APL fu la Video Brain. Secondo il responsabile per i prodotti di questa azienda l'APL presenta numerosi vantaggi sul Basic, primo fra tutti la migliore struttura. Ciò rende il programma più facile da scrivere, da leggere e da mettere a punto.

Anche la Microsoft condivide l'opinione dell'esponente della Video Brain, riconoscendo tuttavia che l'APL, forse per il suo aspetto esoterico, viene raramente scelto dai debuttanti in programmazione. Un altro fabbricante di calcolatori individuali, la Vanguard Systems, ha fatto un passo in più introducendo un compilatore APL basato sullo Z80.

WD/90
Pascal MICROENGINE™

PASCAL MICROENGINE™

- 64 K Byte RAM
- Floppy Controller per 4 Drive
  singola/doppia faccia
  singola/doppia densità
- 2 Interfacce seriali
- 2 Interfacce parallele
- Usa lo stesso "Chip-set" dell' LSI 11*
- Microprogrammato per eseguire direttamente istruzioni in PASCAL
- **Prezzo 10 up:**
  sistema completo di software: 2.950.000 Lit. più IVA
  scheda logica: 1.950.000 Lit. più IVA

*LSI è un marchio registrato DEC

COMPREL

COMPREL s.r.l.

20092 CINISELLO B. (MI) - VIALE ROMAGNA, 1
(02) 6120641/2/3/4/5 - Telex: 332484 COMPRL I

Uffici regionali:

40137 BOLOGNA - P.za Azzarita, 6 - Tel. (051) 551306
50127 FIRENZE - Via T. Mabellini, 4 - Tel. (055) 412018
16033 LAVAGNA (GE) - P.za Marini, 20/10 - Tel. (0185) 301100
60025 LORETO (AN) - Via Dante Alighieri, 26/B - Tel. (071) 977693
35100 PADOVA - Via R. De Visiani, 17 - Tel. (049) 750741
00141 ROMA - Via Muzio Clementi, 58/5 - Tel. (06) 3603463 - 360097
10144 TORINO - Via G. Fagnano, 10 - Tel. (011) 472789

Un nome nuovo: Rexon Business Machines

Due personaggi in passato messi in luce nel settore dell'informatica con indovinate iniziative industriali: C. Wang (fondatore della Wangco, una azienda di dischi e controller, ora del Gruppo Perkin Elmer) e F. Zinsli (cofondatore della Computer Machinary, ditta di periferiche) si sono messi insieme ed hanno dato vita alla Rexon Business Machines Inc. attraverso la quale entrare sul promettente mercato dei piccoli sistemi gestionali. Se le cose andranno bene, dicono, nel giro di cinque anni l'azienda potrà evidenziare su un giro di affari di 100 milioni di $. Il prodotto col quale i due contano per poter effettuare la scalata è un sistema basato su un processor centrale costruito sull'Intel 8086 e con capacità di interfacciare fino ad otto o più terminali. Il suo prezzo, nella configurazione normale, sarà di circa 25000 dollari e fra i sistemi con i quali si confronterà ci saranno il Sistema/34 dell'IBM ed il 380 della Burroughs. Oltre che in alcuni aspetti tecnici, il sistema della Rexon si differenzierà da quelli citati e da altri anche nelle modalità di vendita, che sarà attuata, tra l'altro, attraverso computer stores.

Il TRS-80 1° in graduatoria

Nel 1978 l'industria del personal computer (definito come tale un sistema tipo desktop progettato per soddisfare esigenze professionali, commerciali e casalinghe, venduto ad un prezzo non superiore a 15.000 dollari) avrebbe fruttato introiti per circa 500 milioni di $ ed evidenziato una potenzialità di affari per il 1982 di oltre 2,4 miliardi di $.
Sono stime contenute in una indagine da poco ultimata ad opera della Dataquest di Menlo Park in California. Il TRS-80 della Tandy è stato il sistema più venduto: circa 100 mila pezzi, pari al 50% di tutto il mercato in volume e al 21% del mercato in valore. Per numero di calcolatori personali forniti al secondo posto la Dataquest ha classificato i PET della Commodore (25.000 unità), al terzo gli Apple II dell'ominimo fabbricante (20.000). Indi seguono Imsai e IBM con 5.000 unità, Hewlett-Packard con 4.000, Wang e MITS/Pertec con 3.000.
Oltre a valutazioni quantitative, lo studio ha analizzato anche gli aspetti tecnologici del fenomeno: in futuro, questa una delle conclusioni, un numero sempre maggiore di periferiche verrà ad essere integrato nel chassis centrale e sempre più potenza di calcolo si renderà disponibile ad ogni dollaro aggiuntivo di spesa.

Un milione di piccoli sistemi gestionali

Nel 1987 si consumeranno negli USA un milone di piccoli sistemi gestionali per un valore di 22 miliardi di $. Lo afferma la Frost & Sullivan, nota azienda americana operante nel settore del marketing e delle ricerche. Fino a qualche anno fa solamente le aziende con un giro di affari superiore ai 20 milioni di $ avevano convenienza a ricorrere alla computerizzazione dei loro servizi. Con gli sviluppi nella tecnologia elettronica ora anche le aziende con un fatturato annuo inferiore al milione di $ possono vantaggiosamente ricorrere al calcolatore dimensionando ovviamente l'investimento. Una situazione altrettanto ottimistica esiste anche per il mercato europeo.

Le prime vittime del personal computing

Nell'ambito del settore dei calcolatori personali non sono stati pochi i casi di fabbricanti che si sono sviluppati investendo gli anticipi di pagamento dei clienti. Per questo motivo, o comunque per altri motivi di sottocapitalizzazione, alle prime difficoltà commerciali alcune aziende non hanno saputo reagire, candidandosi alla bancarotta. Per mancanza di liquidità sono entrate in crisi la Imsai Mfg, la Polymorphic Systems e la Digital Group. Altre aziende hanno invece evitato di investire in nuovi prodotti per evitare passi falsi lasciando uno spazio maggiore ai costruttori finanziariamente più dotati.

Partono le consegne dello Z-8000

Chi volesse provare uno Z-8000 può, rivolgendosi alla Zelco s.r.l. di Milano (rappresentante per l'Italia della Zilog), ottenere la consegna del nuovo potente microprocessore da 16 bit, nei normali tempi di consegna e ad un prezzo di Lire 180.000. Analoga annotazione vale per lo Z8 il quale viene fornito ad un costo di L. 96.800 (per grossi quantitativi i prezzi ovviamente scendono). In un comunicato la Zelco fa presente che:

- tutti i componenti sono a stock (compresi quelli come i SIO che altri non forniscono ancora);
- il prezzo della CPU in package plastico per 100 pezzi è di Lire 10.400, mentre quello per i PIO e CTC è di L. 7.200 (per quantità i prezzi sono ovviamente ancora più bassi);
- la capacità produttiva della Zilog viene ora ad aumentare ancora con l'entrata in funzione del nuovo complesso dell'Idaho e di Manila; ciò che lascia prevedere prezzi ancora più interessanti.

GM: 27.000 μP al giorno !

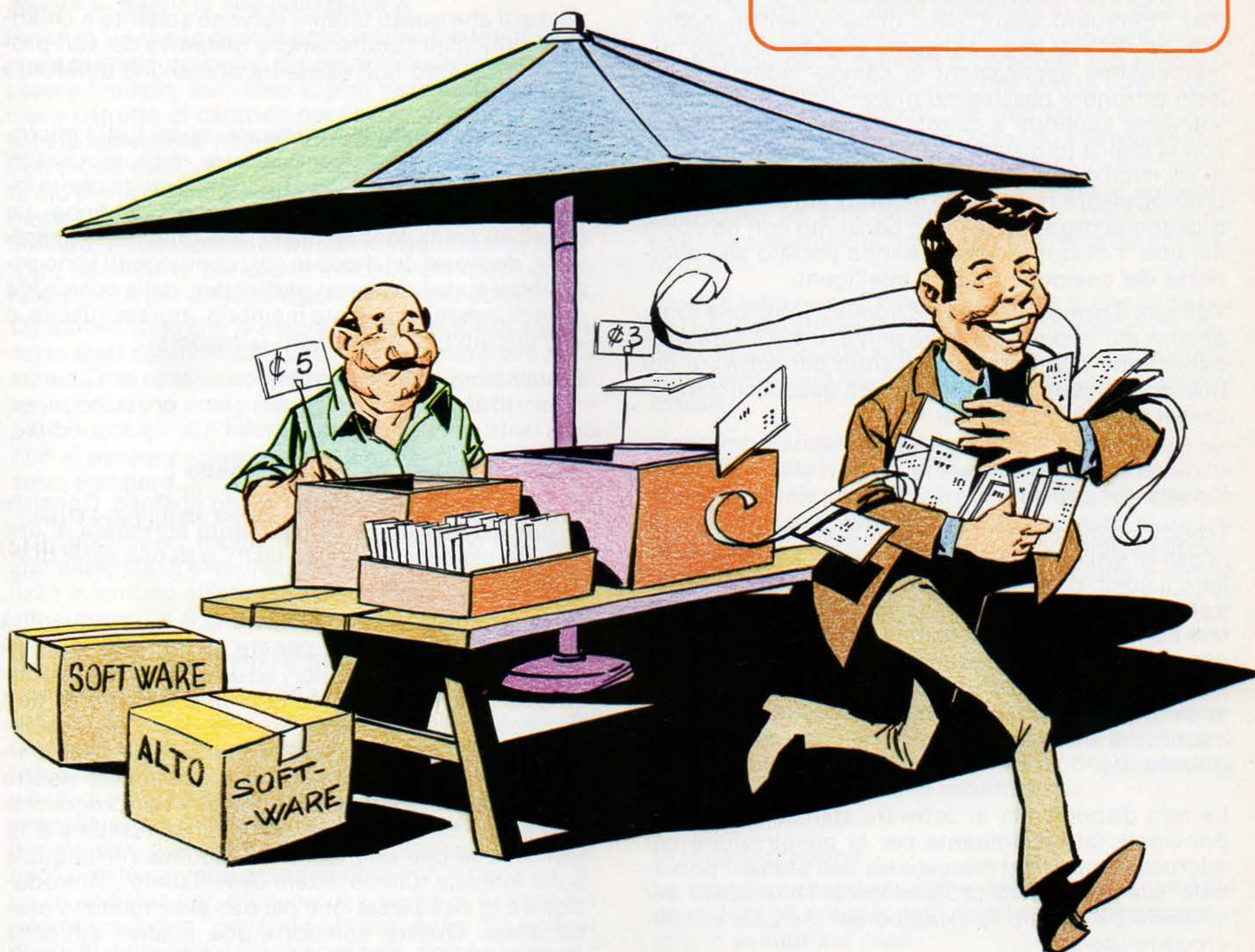
Fra un paio di anni la General Motors potrebbe in capo ad ogni giorno lavorativo consumare 27.000 microprocessori.
E' una stima che viene fatta senza che dall'interno della casa automobilistica venga una qualche indicazione di conferma. Si sà però che la Delco, la divisione per l'elettrotecnica della GM, sta dandosi parecchio da fare. Corrono molte voci di trattative tra la Fairchild e la Ford in vista di un accordo di fornitura: se per circuiti di nuovo tipo o di seconda sorgente rispetto a quelli della Motorola (già legata alla Ford da impegni) non si sa.

L'ENIAC e l'F8 a raffronto

Appena trent'anni separano il primo calcolatore numerico a tubi, l'ENIAC, da un microcalcolatore come può essere l'F8 della Fairchild. Le principali caratteristiche dei due sistemi vengono messi a raffronto nel sottostante quadro con note di commento allo stesso dovute alla Fairchild.

Parametri	ENIAC	F8	Considerazioni
Taglia	85 m^3	0,31	300 mila volte più piccolo
Consumo	140KW	2,5W	56 mila volte in meno
ROM	16Kbits (x)	16Kbits	
RAM	1Kbits (xx)	16Kbits	8 volte più potente
Orologio	100 kHz	2MHz	20 volte più rapido
Transistori o tubi	18.000 tubi	20.000 transistori	
Resistenze	70.000	nessuno	nell'F8 le resistenze sono realizzate con gli stessi componenti attivi
Condensatori	10.000	2	5.000 volte in meno
Relé e commutatori	7.500	nessuno	
MTBT	in ore	in anni	almeno 10 mila volte meglio
Peso	30 tonnelate	500 gr. circa	

(x) a relé e commutatori
(xx) con accumulatori glip-flop

NUCLEO

Software a basso costo: nuova realtà!

di **Rodnay Zaks** - *Sybex*

L'estensione del mercato dei microcomputers, il riferimento a linguaggi di programmazione ad alto livello standard sono in grado di cambiare una linea di tendenza che finora è stata un limite nell'utilizzo di questi sistemi.

Introduzione

Il prezzo dell'hardware dei microcomputer si è ridotto al punto che un componente quale il microprocessore costa, in grandi quantità, attorno alle 1000 lire; si è imposta una standardizzazione di fatto per la maggior parte dei microcomputers con il cosiddetto bus S-100; sono apparsi due interpreti standard BASIC (il C-BASIC ed il BASIC della Microsoft); infine si è avuta la vendita di decine di migliaia di microcomputers indirizzati ad una utenza "personale" e professionale: questi elementi permettono di affermare che per la prima volta nella storia dell'informatica si sono realizzate le condizioni per un mercato di massa del software.

E' avviata una nuova rivoluzione: avremo il software a prezzi paragonabili a quelli dell'hardware, cioé ad un costo molto basso. Scopo di questo articolo è esaminare la natura, le origini e le conseguenze di questo fenomeno nuovo ed importante.

La seconda rivoluzione dei microcomputers

E' stato detto che i microcomputers hanno provocato una nuova rivoluzione industriale. In effetti, in virtù del loro costo notevolmente basso (per grandi quantità), i microprocessori sono ormai inseriti in moltissimi prodotti di largo consumo ed hanno avuto numerosissime applicazioni in campo industriale: di fatto offrono a bassissimo prezzo prestazioni notevolmente superiori a quanto si riusciva a realizzare con la logica tradizionale. Il costo in grandi quantità di un microprocessore è basso, di modo che un microprocessore può essere incorporato si può dire in qualunque prodotto di largo consumo con un costo minimo. I microprocessori hanno portato alla comparsa dei cosidetti prodotti intelligenti.

Tuttavia è ormai avviata una nuova rivoluzione forse ancora più importante della prima. Il 1979 sarà probabilmente ricordato come l'anno del software dei microcomputers. Esaminiamo ora questa circostanza nei suoi diversi aspetti.

Il costo del software

Tradizionalmente in tutti i sistemi di elaborazioni dati i costi di sviluppo del software hanno sempre costituito il costo dominante dello sviluppo del prodotto, tranne che per i sistemi più semplici. I microcomputers non hanno fatto eccezione alla regola, ed anche per essi il costo di programmazione è elevato: anzi si rileva che, in corrispondenza con il costante aumento del costo della mano d'opera, i costi di programmazione di un microcomputer continuano a salire regolarmente.

La non disponibilità di software standard è stata il principale fattore frenante per la penetrazione dei microcomputers nel mercato sia dell'utenza "personale" sia dell'utenza professionale: *l'alto costo del software ha frenato lo sviluppo del mercato dei microcomputers.*

Nelle applicazioni industriali, in quei settori per i quali si è individuato un mercato di decine o centinaia di migliaia di unità, quale ad esempio quello dell'elettronica civile (televisori a colori, prodotti elettrodomestici), diventa economico sviluppare un software specifico per il tipo di applicazione: il costo di sviluppo del software è ripartito su un altissimo numero di unità e si riduce ad un costo unitario basso.

Tuttavia, nel caso di un microcomputer "general purpose", il software di cui deve essere fornito è in generale molto più complesso di quello richiesto ad esempio in semplici applicazioni industriali; pertanto è necessario vendere un gran numero di prodotti di questo tipo per ottenere bassi prezzi di costo.

Probabilmente occorre che il mercato assorba come minimo una quantità attorno alle decine di migliaia di unità perché diventi economico produrre e vendere software per microcomputers: *di fatto un mercato di tale consistenza è ormai presente.*

Per comprendere come ciò è avvenuto, occorre innanzitutto fare riferimento ai differenti tipi di hardware esistenti.

Si possono raggruppare i diversi microcomputers in tre categorie principali: *microcomputers su singola scheda, microcomputers integrati e microcomputers "general purpose".*

(Si badi che questi termini servono soltanto a chiarire le principali caratteristiche distintive dei vari prodotti, in quanto non esiste fra di essi una differenza assoluta).

Dal punto di vista dell'hardware, quasi tutti i microcomputers utilizzano uno dei tre microprocessori principali (l'8080, lo Z-80 e il 6502), di modo che le loro prestazioni sono essenzialmente identiche. Le principali limitazioni tecniche, determinate dall'hardware, derivano dal modo in cui i componenti sono assemblati sulla scheda: in particolare, dalla possibilità di aggiungere facilmente memoria, ingressi-uscite, o altri circuiti che potrebbero richiedersi.

Esaminiamo le tre categorie.

Microcomputers su singola scheda

E' il tipo di microcomputer meno costoso. Consiste di una scheda unica abitualmente corredata di una tastiera esadecimale (a 16 tasti), e di una serie di visualizzatori a LED.

Sulla scheda si trovano inoltre la C.P.A., una quantità minima di memoria (in genere 1K di RAM, e 2K di ROM contenenti il monitor) ed un certo numero di interfacce per la connessione di una stampante, tipo teletype, o di un registratore a cassette o di altri semplici dispositivi di ingresso-uscita. A causa della limitata dimensione della memoria e del numero ridotto di ingressi-uscite, si possono eseguire solo programmi non molto estesi. Inoltre, avendosi una tastiera a 16 tasti invece che una tastiera simbolica (nella quale sono indicate tutte le lettere dell'alfabeto), l'introduzione e la visualizzazione dei dati avvengono in esadecimale. Questo comporta una relativa difficoltà d'impiego e una limitazione dei programmi che è possibile sviluppare; pertanto il microcomputer su singola scheda è un eccellente strumento didattico per imparare ad usare i microprocessori, ma non può essere utilizzato in applicazioni complesse.

Microcomputers integrati

Il gradino successivo consiste nell'utilizzare sempre la stessa scheda, ma corredandola questa vota di una tastiera alfanumerica (con tutte le lettere dell'alfabeto), di una maggiore quantità di memoria, della interfaccia per la visualizzazione dei dati su uno schermo tipo televisione e di almeno una interfaccia standard che permetta di collegare una stampante o un registratore a cassette. Si avrà così un sistema detto microcomputer integrato, che di solito comprende anche l'alimentazione e lo chassis per il supporto delle schede. Essendo la scheda di C.P.U. all'incirca la stessa del microcomputer su singola scheda, la velocità di elaborazione è sempre la medesima. Il costo di questo tipo di sistema parte da un minimo che è di solito il doppio di quello della sola scheda di C.P.U., per arrivare ad un massimo di quattro volte il prezzo della scheda di C.P.U., in funzione dell'hardware aggiunto.

Questo tipo di sistema fornisce il modo più economico per accedere alla potenza di elaborazione di un microcomputer: la programmazione a livello simbolico, resa possibile in questo tipo di sistema dalla presenza di un linguaggio ad alto livello come il BASIC, facilita di molto la sua utilizzazione.

Peraltro, anche in questo caso vi sono dei limiti: il numero di tasti funzione o di controllo sulla tastiera può essere limitato, sul video si può visualizzare un numero ristretto di caratteri per linea o di linee (spesso solo le maiuscole, non le minuscole), infine in genere non si può espandere ulteriormente la memoria. Tutto ciò limita la complessità dei programmi eseguibili. Inoltre, le interfacce sono generalmente in numero limitato, e spesso non si possono aggiungere l'hardware necessario per pilotare floppy-disk, e in generale schede per collegare periferiche più complesse.

Da ultimo, dato che la maggior parte di questi sistemi sono stati costruiti ottimizzando le dimensioni delle schede e le relative connessioni allo scopo di ottenere un prezzo di costo ridotto, sono quasi tutti incompatibili con gli altri microcomputers, per cui fra l'altro non si possono usare schede a bus S-100, che ormai sono standard.

A causa di queste caratteristiche questi sistemi sono i più economici tra quelli che permettono l'accesso alle capacità di elaborazione del computer, ma l'utilizzo è limitato all'ambito "personale".

Microcomputers "general purpose"

Questi sistemi fondamentalmente hanno, oltre alla scheda di C.P.U. una alimentazione potente ed uno chassis contenente numerosi connettori per l'inserimento di schede di memoria (fino a valori come 48K o 64K di RAM) e di schede di interfacciamento per la connessione a qualsivoglia tipo di periferica (video, floppy-disk, disco, etc.). La configurazione di questo tipo di sistema varia grandemente in funzione del tipo di applicazioni; comunque a parità di hardware i microcomputers "general purpose" sono più costosi di quelli della categoria precedente, perché sono più alti i costi dell'alimentazione, della scheda di C.P.U. e dello chassis. In compenso non comportano nessuna restrizione alla eventuale estensione del sistema stesso. Questo è pertanto il tipo di sistema generalmente usato per la maggior parte delle applicazioni gestionali.

La situazione negli U.S.A.

Il riferimento agli U.S.A. è d'obbligo: quasi tutti i prodotti hardware e software presenti attualmente in Europa sono americani, sviluppati in risposta ad una domanda di mercato che in origine era prevalentemente hobbystica. All'inizio (più di tre anni fa), infatti, il mercato era costituito essenzialmente da amatori che erano disposti a passare notti intere a realizzare da sé le schede necessarie per fare un sistema completo, e a creare il relativo software di supporto, in quanto quei sistemi ne erano virtualmente sprovvisti. Oggi non esistono praticamente più hobbysti di questo genere; gli acquirenti dei sistemi sono professionisti, ingegneri, tecnici, uomini d'affari, studenti, in una parola utenti che al momento dell'acquisto si propongono un impiego specifico. Comunque il punto essenziale da non dimenticare è che molte decine di migliaia di sistemi sono stati venduti, per anni, in un'epoca in cui non esisteva praticamente nessun supporto software. Questo fenomeno, forse unico negli annali di un'industria, sta ad indicare che un ampio mercato di utenti potenziali si è formato casualmente, e che esistono ormai diverse centinaia di migliaia di utenti disposti a comprare e/o proporre qualsiasi software standardizzato venga loro proposto e/o richiesto. Infatti, da una parte moltissimi utenti hanno già sviluppato il proprio software applicativo, e così hanno fatto fruttare il loro investimento, dall'altra hanno cominciato a proporre il software sviluppato ad altri potenziali utenti, contribuendo così alla creazione di una nuova industria del software.

Questo fatto nuovo, cioè la creazione di una larga base di utenza software, ha ormai rivoluzionato il mercato dell'informatica in modo radicale: per la prima volta nella storia dei computers, esiste un mercato di massa.

Contemporaneamente il costo dell'hardware si è abbassato in tale misura, come conseguenza della domanda iniziale degli hobbysti, che oggi è possibile acquistare un computer potente al prezzo di un impianto ad alta fedeltà. Inoltre ci sono migliaia di persone esperte in campo hardware e software, e fornite della competenza necessaria per creare il tipo di software applicativo desiderato.

A questo punto si è evidenziato un elemento fondamentale: la necessità che il software destinato ad essere venduto ad un gran numero di utenti fosse standardizzato. Vediamo ora come questa esigenza è stata risolta negli U.S.A. e le indicazioni di prodotti e/o di linea che possono essere tratte.

Standardizzazione

La standardizzazione, come del resto praticamente quasi tutti gli sviluppi nell'industria dei microcomputers, è venuta per caso.

Molti lettori ricorderanno che il famoso bus S-100[1], il bus standardizzato usato ormai dalla maggior parte dei costruttori che utilizzano l'8080 o lo Z-80, fu introdotto dalla MITS (allora piccola società di Albuquerque, nel Nuovo Messico) con l'ALTAIR, il primo microcomputer "general purpose" destinato ad una utenza civile. Il successo fu immediato. Poichè la MITS non poteva fabbricare sistemi in quantità sufficiente, un numero sempre crescente di altri costruttori si mise a costruire schede compatibili con il bus S-100, cominciando con schede di memoria, ma fornendo anche sistemi completi.

Il risultato fu l'introduzione sul mercato di un gran numero di microcomputers e di schede compatibili con il bus S-100: *la standardizzazione iniziale dell'hardware ha contribuito in misura rilevante allo sviluppo di questa nuova industria.*

Altro fattore essenziale è stata l'introduzione del sistema operativo CP/M da parte della Digital Research in California. Pur essendoci altri sistemi operativi, possiamo dire senz'altro che il CP/M è diventato

---

[1] *cfr. "Il bus S-100: uno standard de facto" in BIT n.2*

CP/M è un sistema operativo sviluppato per essere utilizzato con microcomputers 8080 e Z-80; è stato impiegato da oltre quattro anni da vari costruttori di sistemi, per cui si può considerare ampiamente collaudato. Le funzioni principali di questo package software comprendono la gestione di files cosiddetti dinamici, l'editing dei programmi, l'assemblaggio, il debugging, l'elaborazione in batch, il caricamento di programmi: di notevole importanza è il fatto che questo package è "aperto", per cui si può facilmente adattare a tutti i microcomputers 8080 o Z-80 con almeno 16k di memoria RAM e uno o due floppy-disk con diskette IBM-compatibili. Viene fornito infatti con un'ampia documentazione per poterlo adattare alle varie configurazioni hardware. Il suo costo è attorno ai cento dollari.
CP/M è stato sviluppato da: DIGITAL RESEARCH Post Office Box 579 - Pacific Grove, California 93950.

quello più usato sui microcomputers. Può essere eseguito su 8080 e su Z-80.

Il vantaggio essenziale è che tutti i files e quindi i programmi sviluppati sotto CP/M sono compatibili fra loro, cosicchè l'utente può creare dei files e tra sportarli su un altro sistema, oppure comprare dei programmi registrati nel formato CP/M, che potranno essere eseguiti immediatamente sul proprio sistema: *la standardizzazione di fatto dei formati dei files permette ormai lo scambio e la distribuzione dei programmi su vasta scala per tutti i possessori di un 8080 o di uno Z-80.*

Da ultimo, si è verificata la convergenza di fatto su un linguaggio di programmazione: il linguaggio BASIC. Oggi due versioni di BASIC, il BASIC della MICROSOFT e il C-basic (BASIC commerciale) sono diventati virtualmente gli standard (per il momento).

Quali che siano i meriti di questi interpreti BASIC, essi forniscono ormai uno standard. Qualunque programma scritto nel BASIC della MICROSOFT o in C-BASIC potrà essere eseguito su qualsivoglia altro sistema fornito dello stesso interprete. Il risultato è stata una esplosione del software, e in particolare del software gestionale, che è ora agli esordi. In *In sintesi, la maggior parte della applicazioni gestionali del microcomputers è ormai eseguita su un microcomputers (utilizzante l'8080 o lo Z-80) che lavora in C-BASIC ed è sotto il controllo del sistema operativo CP/M.*

Tutte le condizioni richieste per un software a basso costo sono ormai realizzate: una domanda considerevole da parte dell'utenza ed una standardizzazione di fatto. Di conseguenza prodotti software di costo notevole possono ormai essere sviluppati e venduti in grandi quantità a basso prezzo.

Il software gestionale

È bene ricordare che il costo di sviluppo del software gestionale è estremamente elevato. Lo sviluppo di buoni programmi, come conti-clienti, conti-creditori, inventario, accettazione degli ordini, etc., è molto alto. Lo sviluppo di questi programmi è notevolmente più costoso, nella maggior parte dei casi, di quello relativo alle applicazioni industriali. Per questa limitazione, ma anche per altri fattori, l'utilizzo dei computers finora è stato ristretto alle grandi società, che potevano permettersi l'acquisto di sistemi costosi e di un software ancora più costoso, nonchè l'impiego continuo di programmatori per la creazione e la manutenzione del software necessario.

Ora nell'industria del software sta per verificarsi un cambiamento analogo all'introduzione del prêt-à-porter nel mondo della confezione, nel senso che sono disponibili moltissimi programmi standard a basso prezzo: *questi programmi non sono l'ideale per ogni singolo utente, ma sono sufficienti per la maggior parte di essi.* E' evidente infatti che nessun programma standard gestionale corrisponderà esattamente alle esigenze di ogni azienda, pur tuttavia la stragrande maggioranza delle piccole e medie imprese avrà modo di beneficiare in misura più o meno estesa dei vantaggi dell'automatizzazione delle procedure gestionali.

Fino ad ora un problema fondamentale per questo tipo di software era il fatto di dover organizzare in modo diverso le stesse informazioni per utilizzare ciascun programma. Ora questo problema è in via di risoluzione, e cominciano ad affacciarsi sul mercato, a basso prezzo, dei packages gestionali integrati di semplice utilizzo. Pertanto è probabile che fra un anno o due il prezzo del software gestionale in particolare, ed in generale del software, sarà diminuito in modo praticamente analogo al costo dell'hardware.

È la prima volta che un fenomeno del genere si verifica nell'industria informatica: infatti tradizionalmente il costo del software è sempre salito, mentre quello dell'hardware scendeva. Oggi, come conseguenza della nuova struttura del mercato, la curva dei prezzi del software a livello di grosso pubblico si appresta probabilmente a seguire quella dei componenti: esponenzialmente verso il basso. Pertanto non c'è più nessuna ragione perchè un'azienda o un professionista debbano avere dei dubbi sulla possibilità di trarre vantaggi da questa nuova situazione.

Conclusioni

È difficile in questa fase fare una stima dell'impatto di questi nuovi sviluppi, dal momento che si è appena agli inizi. Comunque si possono prevedere alcuni punti essenziali. Praticamente tutte le piccole e medie aziende, nonchè i professionisti, avranno ben presto a loro disposizione calcolatori di basso costo capaci di venire incontro alle loro esigenze gestionali, e questo, a sua volta, implicherà cambiamenti di organizzazione nell'ambito dell'azienda stessa e di ruoli delle singole persone: un gran numero di funzioni interne all'azienda saranno modificate, certe attività spariranno, mentre ne saranno create delle altre. In breve, *l'utilizzazione di questi nuovi strumenti modificherà probabilmente la struttura interna dell'impresa.*

La rivoluzione dei microprocessori continua: è ormai in pieno sviluppo nell'industria elettronica grazie ai costi estremamente bassi dell'hardware, comincia adesso all'interno delle medie-piccole aziende con il fenomeno nuovo della disponibilità di software a costo molto basso. Questa circostanza si farà sentire in molti settori della produzione e dei servizi, eliminando molte mansioni, cambiando profondamente un gran numero di procedure, e migliorando in misura notevole la competitività di quelle imprese in cui la transizione sia stata realizzata abbastanza rapidamente. Infine, questa rivoluzione coinvolgerà probabilmente aree di utenza molto più vaste *una volta che si sia creata una rete di distribuzione della informazione.* ■

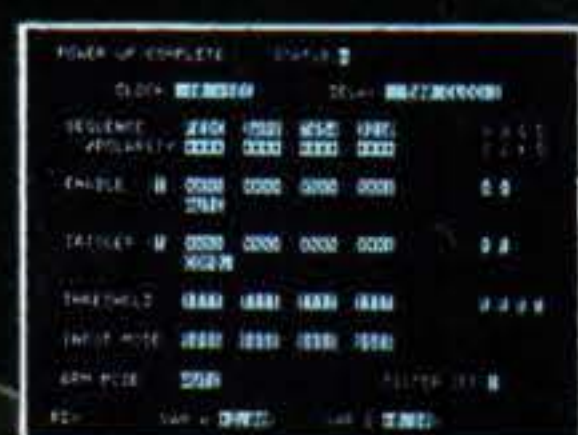
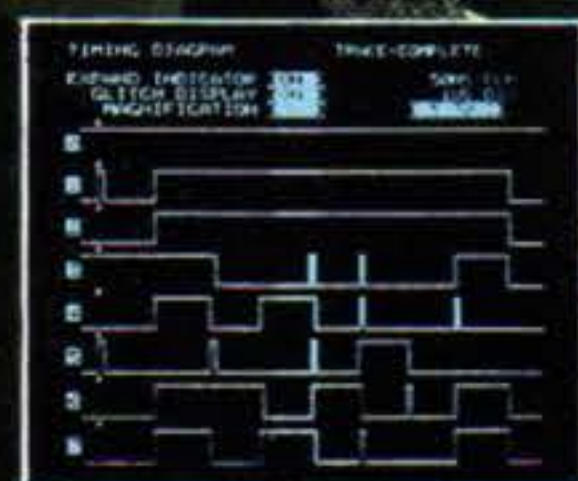
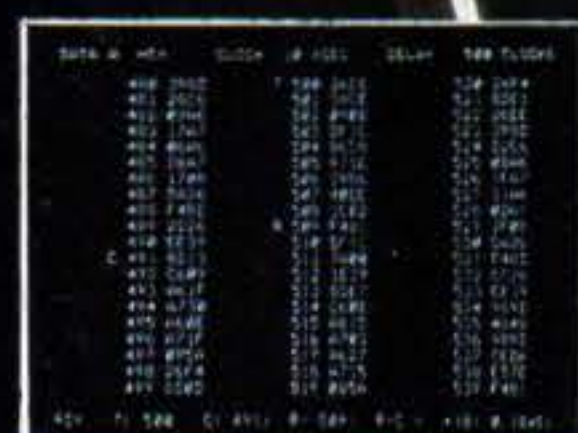
GOULD biomation

la più vasta gamma di logic analyzer

Il K100-D offre anche la rappresentazione dei dati Special Mode, molto utile per controllare l'Interface Bus IEEE 488.

Schermo incorporato con rappresentazione in funzione del tempo e dei dati.

Funzionamento asincrono fino a 100 MHz.

tastiera gestita da μP

1. Status Display

2. Timing Display

3. Data Display

1024 bits/can. più altrettanti per comparazioni

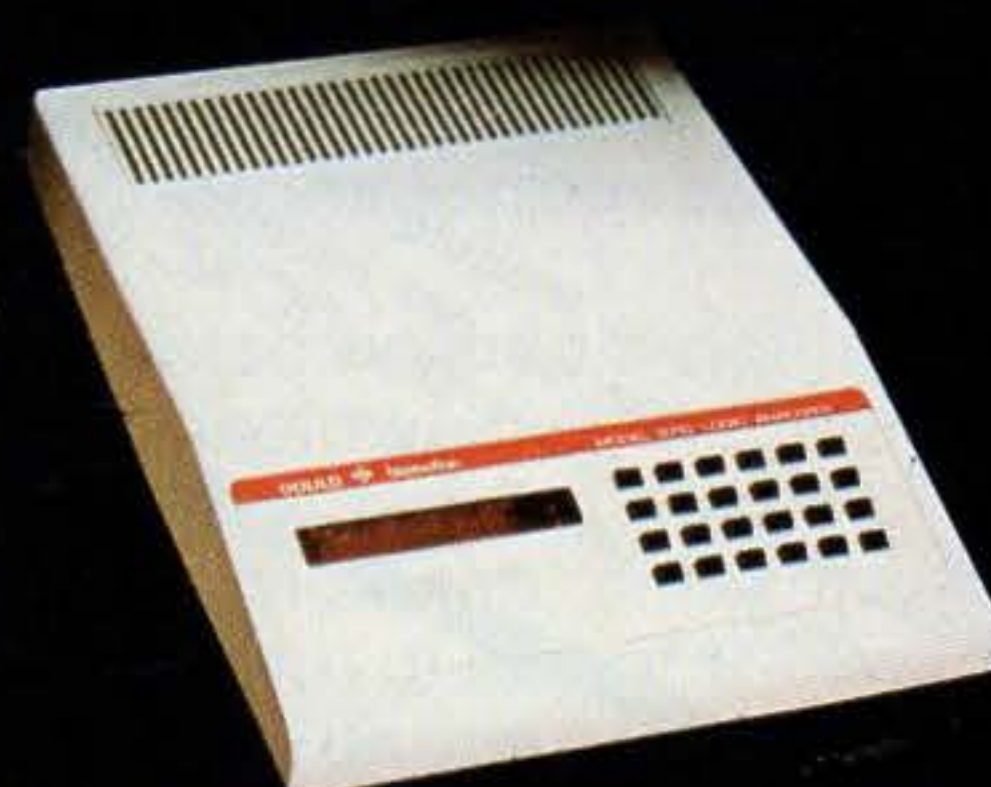
Sonde attive ad alta impedenza realizzate appositamente per il K100-D.

16 canali d'ingresso espandibili a 32 per risolvere anche particolari problemi connessi ai μP.

NUOVO Il Logic Analyzer K100-D è 5 volte più veloce e ha una memoria 4 volte maggiore del modello concorrente più direttamente comparabile. Non a caso questo prodotto è stato realizzato dalla Gould Biomation, la più avanzata costruttrice mondiale di tali apparecchiature. Il K100-D, con la sua possibilità di funzionamento asincrono fino a 100 MHz, i 16 canali di ingresso espandibili a 32, la memoria di 1024 bits per canale, il controllo da tastiera gestito da microprocessore, lo schermo incorporato, la rappresentazione in funzione del tempo e dei dati, il modo di funzionamento Latch selezionabile canale per canale e le sue esclusive sonde attive ad alta impedenza, si può senz'altro definire il più veloce, potente e versatile Logic Analyzer a 16 canali oggi esistente.

Modello	Freq. (MHz)	Latch (ns)	Canali	Memoria
920-D	20	10	9	256 x 9
851-D	50	5	8	512 x 8
1650-D	50	5	16	512 x 16
2710-D	10	-	27	64 x 27
8100-D	100	3	8	2048 x 8
K100-D	100	5	16 + 16	1024 x 16
9100-D	100	5	9	1024 x 9
8200-D	200	1	8	2048 x 8
168-D	10	-	1	256 x 25
DTO-1	Digital Testing Oscilloscope: 1. Logic Analyzer - 2. Storage Oscilloscope 3. Go/No-Go Comparator			

Il modello 920-D è il più economico Logic Analyzer a 20 MHz, 9 canali, 256 bits per canale, latch mode, trigger combinatoriale su tutti i canali ed è particolarmente adatto per il SERVICE dei circuiti digitali.

L. 1.850.000* - consegna pronta

NUOVO Il modello 2710-D è stato realizzato principalmente per l'analisi dei circuiti a microprocessori. Ha 27 canali di ingresso, 10 MHz, 64 bits per canale.

L. 1.850.000* - consegna pronta

una gamma completa di strumenti elettronici di misura

elettronucleonica s.p.a.

MILANO - Piazza De Angeli, 7 - tel. (02) 49.82.451
ROMA - Via G. Segato, 31 - tel. (06) 51.39.455

Bit

elettronucleonica S.p.A.

Desidero

☐ ricevere maggiori informazioni sul Logic Analyzer Gould Biomation modello ....

☐ avere una dimostrazione del Logic Analyzer Gould Biomation modello ....

Nome e Cognome ....

Ditta o Ente ....

Indirizzo ....

* Nov. 78 - Pag. alla consegna, IVA esclusa

Sistemi gestionali realizzati con microcomputer

di **Luigi Mutti** - *Omega Data S.r.l. - Milano*

I microcomputers, dato il loro basso costo e la flessibilità di applicazioni, offrono interessanti prospettive nel campo dell'automazione in generale, sia a livello industriale che a livello amatoriale. L'idea di utilizzarli anche per l'applicazione più classica, quella gestionale, è indubbiamente affascinante . . .

Introduzione

Tra i campi di applicazione della tecnologia dei microcalcolatori, quello dei Sistemi di Gestione Aziendale è tra i più promettenti, anche se è tra quelli che meno si prestano alle improvvisazioni.
Questo articolo e gli altri che seguiranno sul medesimo argomento vogliono dare una panoramica, certamente non esaustiva, del mercato gestionale nella fascia di utenza che può essere servita efficientemente da sistemi realizzati con microcalcolatori.

Perchè il microelaboratore

La struttura industriale del nostro Paese, formata principalmente da Aziende di piccola e media dimensione, ha sempre favorito ditte operanti nel settore EDP in grado di fornire sistemi piccoli e, soprattutto, di semplice impiego.
In questo contesto, l'avvento dei minicomputers negli anni '60 ha dato luogo ad un fiorire di Case fornitrici di calcolatori in grado di soddisfare le esigenze della fascia media del mercato EDP; tuttavia il costo ancora troppo elevato dei sistemi basati su questi minicomputers e la situazione generale del Paese, che non incoraggiava investimenti, ostacolavano la diffusione degli elaboratori elettronici, che rimanevano perciò degli strumenti di calcolo d'avanguardia e d'élite.

Un grosso passo era però già stato fatto: la figura tradizionale del grosso elaboratore, inavvicinabile e mostruoso nella sua potenza, cominciava a vacillare nella mentalità comune; grazie alle nuove tecnologie, i minicomputers non richedevano ambienti termicamente condizionati, e potevano essere inseriti nel contesto aziendale come una normale macchina elettrocontabile; soprattutto, dato il basso costo di utenza, non imponevano lo sfruttamento totale del tempo macchina, che a sua volta comportava una rigida preorganizzazione del lavoro e un certo numero di persone adibite esclusivamente alla perforazione dei dati con cui alimentare ininterrottamente la macchina.
Ma la spinta decisiva verso un'ampia diffusione dei sistemi elettronici di calcolo veniva, negli anni '70, dall'entrata dei microcomputers sul mercato.
Questi nuovi prodotti tecnologici, che consentono l'implementazione di sistemi di elaborazione in modo relativamente semplice - da una parte -, e la messa a punto di nuovi dispositivi periferici a basso costo (quali, ad esempio, i floppy-disk e le stampanti a matrice) - dall'altra -, offrono oggi la possibilità di realizzare strumenti di calcolo efficienti a prezzi decisamente impensabili fino a pochi anni fa.

Inoltre, la ripresa degli investimenti da parte delle piccole e medie industrie, volti a razionalizzare la gestione aziendale, e il notevole appesantimento degli adempimenti fiscali, contribuiscono ulteriormente alla diffusione massiccia degli elaboratori elettronici, e, in particolare modo, dei microelaboratori.

Dimensionamento del sistema

Per ottenere un buon sistema gestionale occorrono:

- un hardware efficiente;

- un software di base appositamente studiato per applicazioni EDP;
- un software applicativo con una Biblioteca di programmi standard completa e facilmente adattabile alle esigenze dell'utente.

L'hardware di questi sistemi, in generale, è equipaggiato con una Unità Centrale dotata di una memoria di lavoro da 32 a 64 Kilobytes; una Memoria di Massa basata su due o quattro Floppy-disk a singola o doppia testina, singola o doppia densità (per una capacità complessiva di immagazzinamento dati da 500.000 a 4.000.000 bytes).
Se fosse necessaria una capacità di archiviazione superiore a quella ottenibile con i floppy-disk, o nel caso il sistema debba servire più terminali ed eseguire parallelamente più di un programma (siamo a livello di multiprogrammazione), si possono vantaggiosamente utilizzare Dischi da 5 o da 10 Megabytes, eventualmente affiancati da una singola unità a floppy-disk da usarsi come unità di immissione di dati preparati fuori linea, o per l'archivazione di una limitata quantità di informazioni di uso saltuario o temporaneo; una stampante, con velocità di stampa variabile tra gli 80 e i 200 caratteri per secondo; infine uno o più terminali video, tramite i quali si può svolgere il dialogo operatore-calcolatore.
Come si può intuire, il limite massimo dell'espansione di un sistema realizzato con microcomputer è più pratico che teorico, ed è dettato in massima parte da considerazioni di tipo economico.
In teoria, infatti, ad un sistema di questo tipo si possono collegare dispositivi periferici molto più potenti di quelli citati sopra, come, ad esempio, più unità a dischi magnetici dalla capacità di archiviazione di 80 Megabytes ciascuno, e stampanti a catena, unitamente a periferiche tradizionali come il lettore-perforatore di schede, ecc. Si potrebbe, in altri termini, realizzare un centro di elaborazione dati strutturalmente simile a quelli della prima generazione.
D'altra parte, un sistema così configurato sarebbe estremamente poco efficiente: il basso costo dell'unità centrale diventerebbe irrilevante rispetto al costo globale, inevitabilmente alto, delle periferiche, e la velocità della CPU, ovviamente inferiore rispetto a quella dei sistemi di più grandi dimensioni, non ne permetterebbero lo sfruttamento ottimale.
Inoltre, approssimandosi la velocità dei dispositivi di Input/Output a quella dell'unità centrale, si verificherebbe lo spiacevole fatto che, nel tempo di un'operazione di I/0, la CPU sarebbe in grado di eseguire solamente poche operazioni, e, probabilmente, nemmeno tutte quelle necessarie a consentire il passaggio tra i diversi programmi in attesa di esecuzione in memoria centrale.
In una condizione di questo genere, si vede facilmente come non sia attuabile la multiprogrammazione, e in definitiva questo sistema, pur dotato di apparecchiature periferiche velocissime e potenti (e costose), non offrirebbe prestazioni altrettanto sofisticate.
Alla luce di queste considerazioni, si capisce quale importanza rivesta l'accurato bilanciamento di tutte le risorse del sistema, e come dispositivi meno potenti, ma calibrati rispetto alle possibilità dell'unità centrale, aumentino sensibilmente l'efficienza globale del sistema, ottimizzandone il rapporto prestazioni/costo.

Il software

Nella globalità dei problemi che concernono la strutturazione e l'implementazione di un sistema elettronico di elaborazione, il discorso del software rappresenta la fase più critica e delicata.

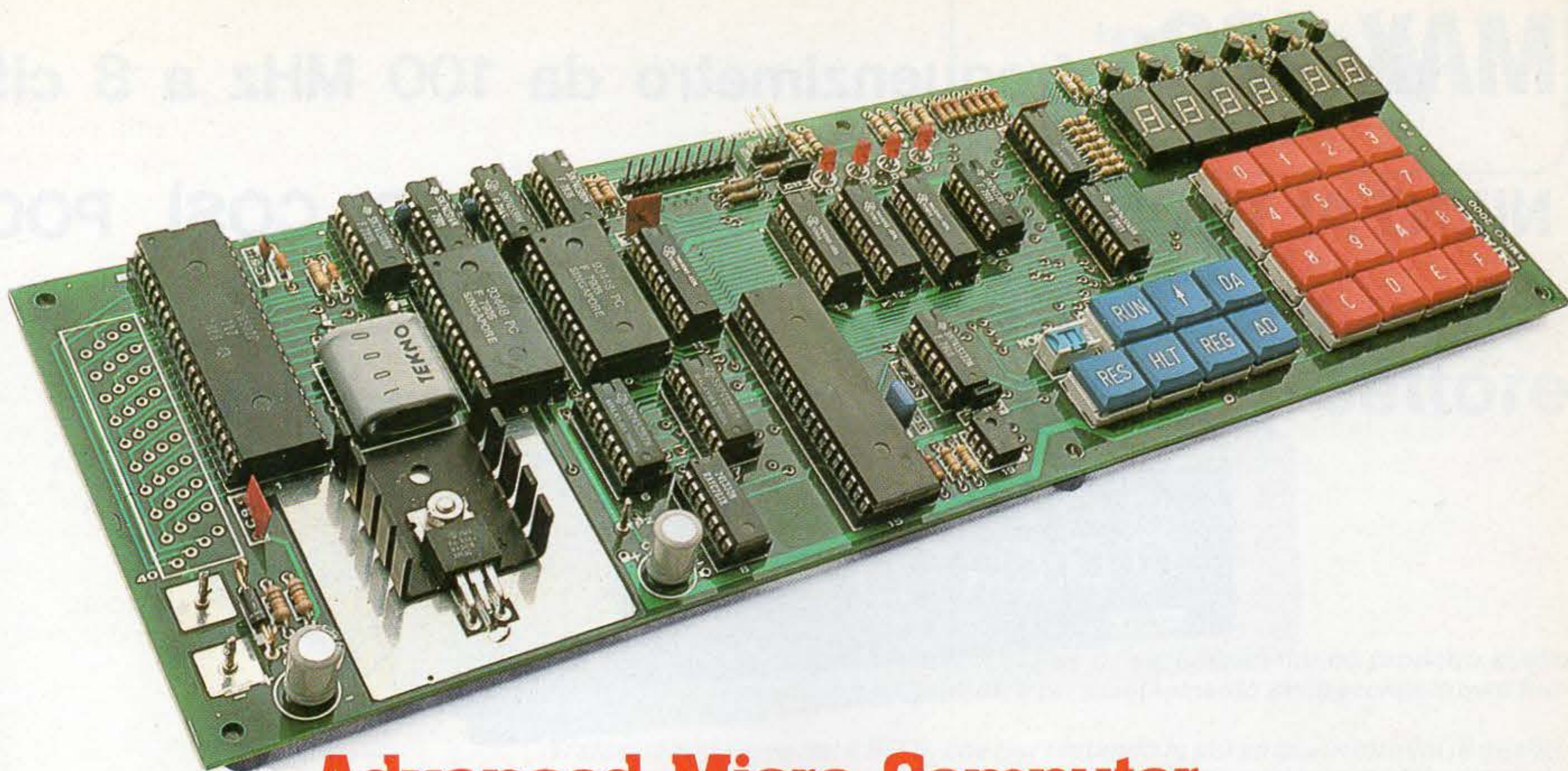
Una volta stabilita in modo efficiente l'architettura dell'hardware, i risultati e la qualità delle prestazioni del sistema di calcolo dipendono quasi esclusivamente dal software disponibile.

In questo senso, il software di base riveste il ruolo più importante, anche se il meno visibile: dalla potenza del sistema operativo dipenderanno, infatti, l'efficienza e la flessibilità di tutti i programmi applicativi, e, soprattutto, il costo di produzione del nuovo software (generalmente sempre piuttosto elevato).

Questo discorso è, comunque, molto ampio ed interessante e merita di essere riaffrontato più dettagliatamente in seguito.

Vantaggi di una soluzione con microcomputer

La scelta di fornire solamente l'hardware della macchina e di demandare la produzione del software a Case specializzate, se costituisce una soluzione vantaggiosa per chi vende il sistema, non è altrettanto consigliabile all'utente del sistema stesso.
Nella scelta di un elaboratore elettronico, infatti, è molto importante valutare, oltre che le prestazioni della macchina, anche le risorse globali di cui l'utente può usufruire.

Spesso chi acquista un minicalcolatore viene messo in condizione di dover trattare separatamente con chi fornisce la macchina, con chi esegue l'analisi dei suoi problemi, e con chi sviluppa i suoi programmi; questa situazione, oltre a creare disagi all'utente, è comunque onerosa in termini di tempo di realizzazione e di investimenti; a volte, la definitiva stesura dell'analisi e dei programmi rivela che il calcolatore già acquistato ha una potenza di elaborazione inferiore a quella necessaria, oppure che una macchina di dimensioni minori potrebbe assolvere gli stessi compiti con costi più bassi.
Questi problemi verrebbero eliminati se lo studio sul tipo e sul dimensionamento dell'elaboratore venisse concretamente integrato con l'analisi delle necessità dell'utente.
I sistemi basati su microcomputer si prestano, grazie alla relativa facilità di implementazione ed al basso costo dei componenti, ad essere realizzati anche da gruppi di progettisti hardware e software che perseguano la politica di offrire all'utente un servizio completo con applicazioni "chiavi in mano"; un servizio, cioè, che comprenda, oltre alla fornitura dell'hardware e alla sua successiva manutenzione, anche l'approntamento e l'aggiornamento delle procedure applicative adatte a risolvere concretamente i problemi del Cliente.
Riassumendo, i vantaggi che possono derivare dalla scelta di un sistema di calcolo basato su un microcomputer rispetto ad un minicomputer sono un hardware flessibile dal costo decisamente basso e di facile manutenzione, e la ricomposizione, di fronte all'utente, di un'unica figura di ditta fornitrice sia della macchina, sia dei programmi. ■

Advanced Micro Computer

AMICO 2000

Il cuore del sistema.

Un sistema completo a microelaboratore da autocostruire e tutto il supporto didattico necessario.

Caratteristiche

CPU: microprocessore 6502
Memoria RAM: 1kbyte
Memoria ROM contenente il Monitor
Tastiera esadecimale + tasti funzionali e passo singolo
Visualizzazione LED a 6 cifre
8 linee di ingresso e uscita parallelo
Generatore di clock quarzato
Regolatore di tensione incorporato
Alimentazione 5Vcc (non regolati), 800mA max.
Predisposto per l'espansione della RAM (1K)
Predisposto per l'interfaccia con registratore a cassette

AMICO 2000 A
è un microelaboratore autonomo
e costituisce la base del sistema
che viene completato
dalle seguenti schede di espansione:

ALIMENTATORE DI POTENZA
ESPANSIONE RAM/EPROM
INTERFACCIA TV
TASTIERA ALFANUMERICA
INGRESSO/USCITA DIGITALE
INGRESSO/USCITA ANALOGICO
CONTENITORE A RACK

Prezzi AMICO 2000A (IVA 14% esclusa)

- In scatola di montaggio Lit. 195.000
- Kit ER1 di espansione 1kByte RAM Lit. 25.000
- Kit EC2 per interfaccia registratore a cassette Lit. 30.000
- Versione montata e collaudata completa di espansione RAM e interfaccia cassette Lit. 285.000

Il sistema AMICO 2000 viene descritto in una serie di articoli didattici pubblicati sulla rivista SPERIMENTARE a partire dal numero di Dicembre 1978.

L'AMICO 2000 è progettato in Italia dalla

A.S.EL. s.r.l. Via Cortina D'Ampezzo, 17 Milano - Tel. 02/5391719

Bit

Prego inviarmi senza alcun impegno da parte mia:

☐ Ulteriori informazioni sul sistema AMICO 2000
☐ Le modalità per l'ordinazione e il pagamento

Nome ______________ Cognome ______________
Via ______________ N. ______
Città ______________ C.A.P. ______

Realtà e prospettive della Fairchild Camera and Instruments nel settore dei microprocessori

di Roberto Stefanelli

Fairchild Semiconduttori S.p.A.

Nel settore dei microprocessori, il tempo e l'esperienza hanno prodotto quello che a nostro avviso si può ben giudicare un assestamento delle architetture e funzioni.

Tralasciando l'area del 4 BITS, che pur restando la più ampia in termini di quantità di prodotto impiegato coinvolge funzioni e architetture relativamente semplici e valori unitari modesti, il nostro gruppo si è orientato in maniera decisa verso le aree dell'8 BITS e del 16 BITS, dense di fermento tecnologico e di grande potenziale quantitativo e di valore. Nel settore a 8 BITS il nostro gruppo è stato a suo tempo antesignano e promotore dell'architettura e funzione "di controllore" per un micro con porte di "input/output" e architettura tale da poter essere impiegato in una molteplicità estremamente vasta di funzioni di controllo con grande flessibilità.
In questa visione concettuale abbiamo prodotto il popolare microprocessore F8 e più recentemente il singolo "CHIP" 3870 e stiamo per produrre la sua versione singolo "CHIP" EPROM 38E70, perfettamente compatibile con lo stesso e con la sua versione maggiorata a 4K di memoria, 3872.

Sempre nel settore 8 BITS il nostro gruppo ha identificato nel 6800 l'architettura e la funzione di processore per elaborazione dati più moderna ed a largo spettro.

Per evitare un' operazione di copiatura di per sè lunga ed imperfetta si è provveduto tramite accordi di licenza incrociata ad acquisire le maschere direttamente dal primo costruttore. Così facendo il nostro gruppo è oggi in condizione di fornire all'industria due famiglie "leaders" nel proprio ambito quali F8/3870 e il 6800 ed è ovvio che si farà ogni sforzo per tenere aggiornate queste due famiglie secondo le possibilità degli sviluppi tecnologici e la richiesta di funzioni di interfaccia. Nel campo del 16 BITS ove i confini tra microprocessore e "minicomputer" tendono ad avvicinarsi rapidamente, il nostro gruppo dispone innanzittutto di una tecnologia bipolare I^3L frutto di lunga e notevole esperienza compiuta con altri prodotti in grado di consentire impaccamenti e velocità operative oggi non emulabili con tecnologie NMOS anche avanzatissime.

Secondariamente, con tale tecnologia il nostro gruppo ha già realizzato il primo CPU di una famiglia commercialmente denominata "Microflame" CPU conosciuto come 9440.

Tale prodotto capostipite è già corredato, e sarà ulteriormente arricchito, di "software" operativo in linguaggi evoluti che nel complesso sono commercialmente denominati come famiglia "fire".

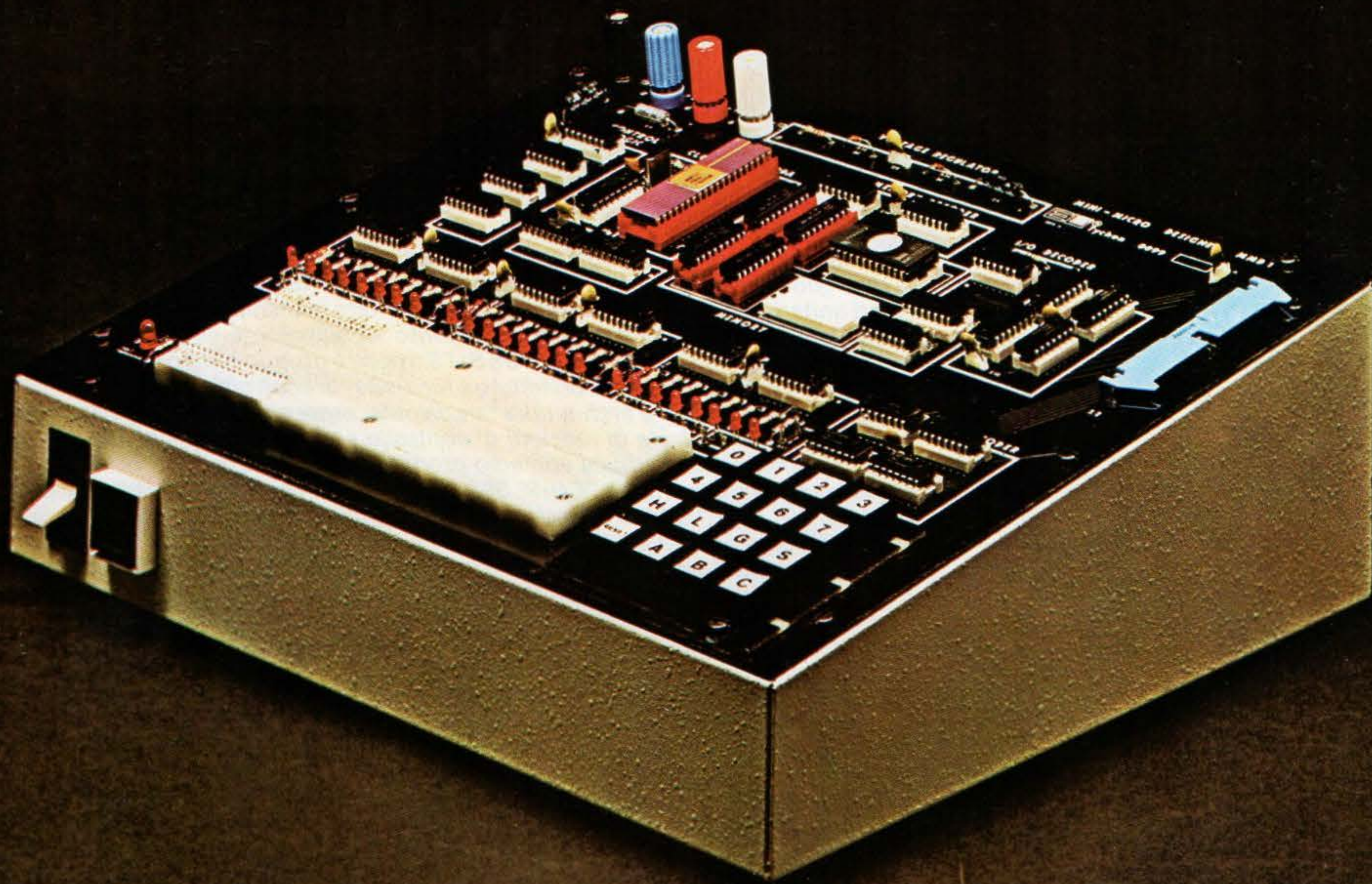
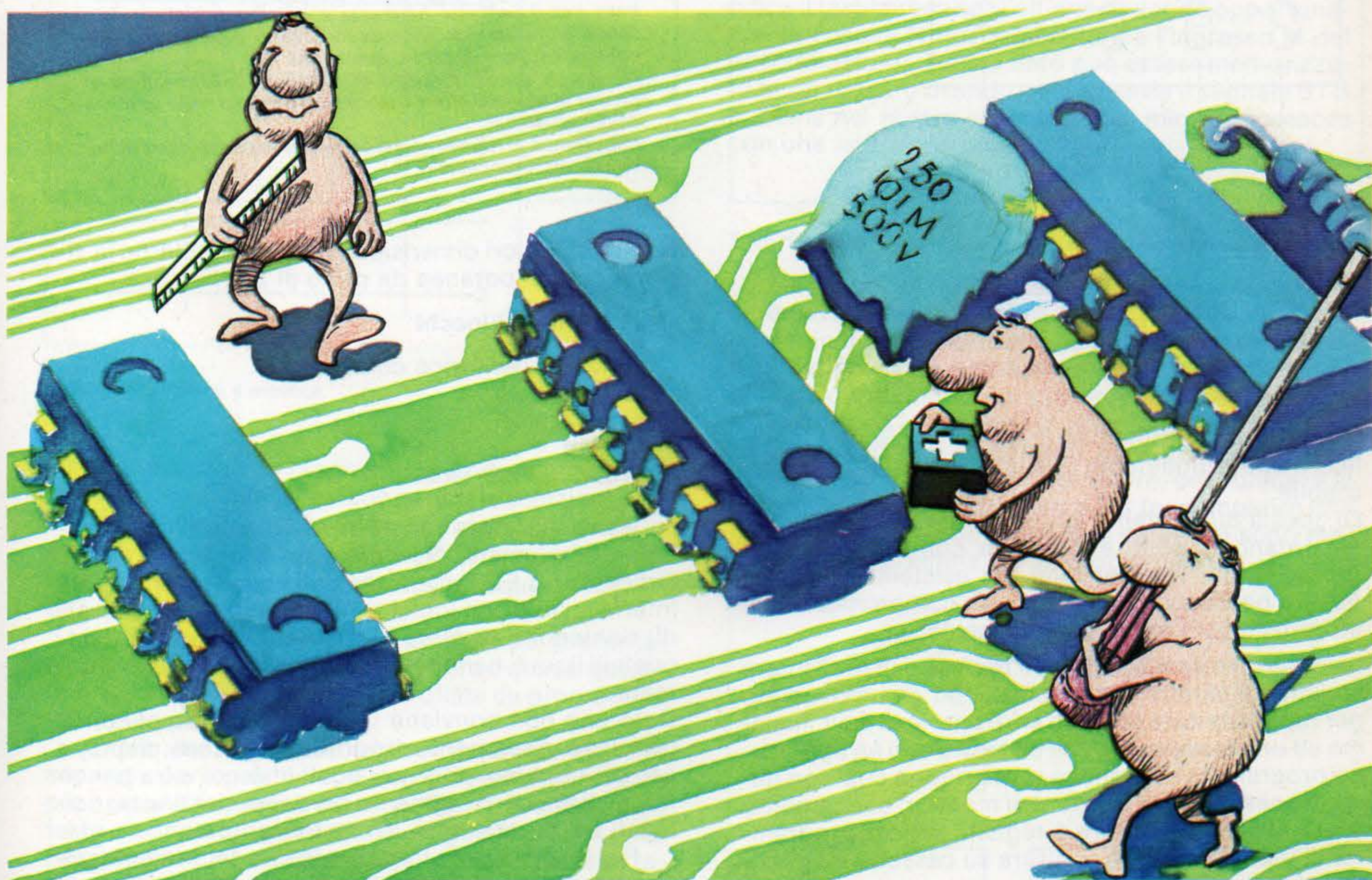
Dal "Basic" al "Assembly" dal "Macro" al "F DOS" fino al "Baby Basic" e ai futuri Pascal ed altri, la famiglia di "software" già sviluppata o in corso di sviluppo sarà direttamente compatibile con i futuri CPU.

Altrettanto è possibile dire per il 9441 ("Memory Control Units") e per il 9442 ("I/O Bus Controller") "Chips" dedicati alle specifiche funzioni descritte. Il 9440 già in produzione a 2192 differenti istruzioni divise in quattro classi e opera con logica statica che richiede un singolo orologio da continua a 12 MHz e naturalmente singola alimentazione a 5 Volt e gamma di temperatura civile e militare.

Il fratello maggiore 9445 denominato "Microflame 2" con un'avanzamento tecnologico dello stesso processo potrà indirizzare direttamente 128.000 "Bytes" e con "Memory Mapping" fino a 16 milioni di "Bytes".

La sua velocità operativa andrà da DC a 15 MHz con 5 Volt di "Power Supply" e gamma di temperatura civile e militare. Il 9445 andrà in produzione alla fine del 1979 e sarà per tanto il secondo microprocessore 16 Bits che il nostro gruppo gestirà in produzione.

Un terzo e più potente prodotto denominato 9450 seguirà i due già descritti con un tempo di introduzione previsto verso la fine del 1980.

In queste condizioni il nostro gruppo disporrà di uno spettro di prodotti nel campo dei microprocessori con pochissima concorrenza per ampiezza globale e nessuna per ampiezza nell'area del 16 Bits.

Volete farvi un "PICOCOMPUTER"?

di D. Del Corso

Cominciate a cercare una calcolatrice tascabile con visualizzatore a LED ed almeno 20 tasti organizzati a matrice. "Curatela" come qui descritto ed avrete un pariferico per il vostro microprocessore.

Il picoperiferico

Un elemento indispensabile per rendere operativo qualunque sistema a microprocessore è un organo per scambiare informazioni col mondo esterno, in altre parole un periferico. Nel caso di periferici classici (teletype, terminale video), il loro costo ha una incidenza significativa sul costo del sistema, arrivando al limite ai "single board computer" del costo di qualche centinaio di migliaia di lire, che richiedono qualche milione di lire di periferici. Questo diventa un grave ostacolo quando si vogliono organizzare molti posti di lavoro operanti su microelaboratori (caso tipico delle scuole) e per chi vuole accostarsi con approccio hobbystico al microprocessore, senza doversi impegnare in spese rilevanti.

Le funzioni minime richieste ad un periferico al fine di permettere l'interazione con l'operatore sono:

GAME TORINO

a cura del Gruppo Amatoriale Micro-Elaboratori, Torino *e del Laboratorio Didattico di Elettronica Applicata dell'Istituto di Elettronica e Telecomunicazioni del Politecnico di Torino.*

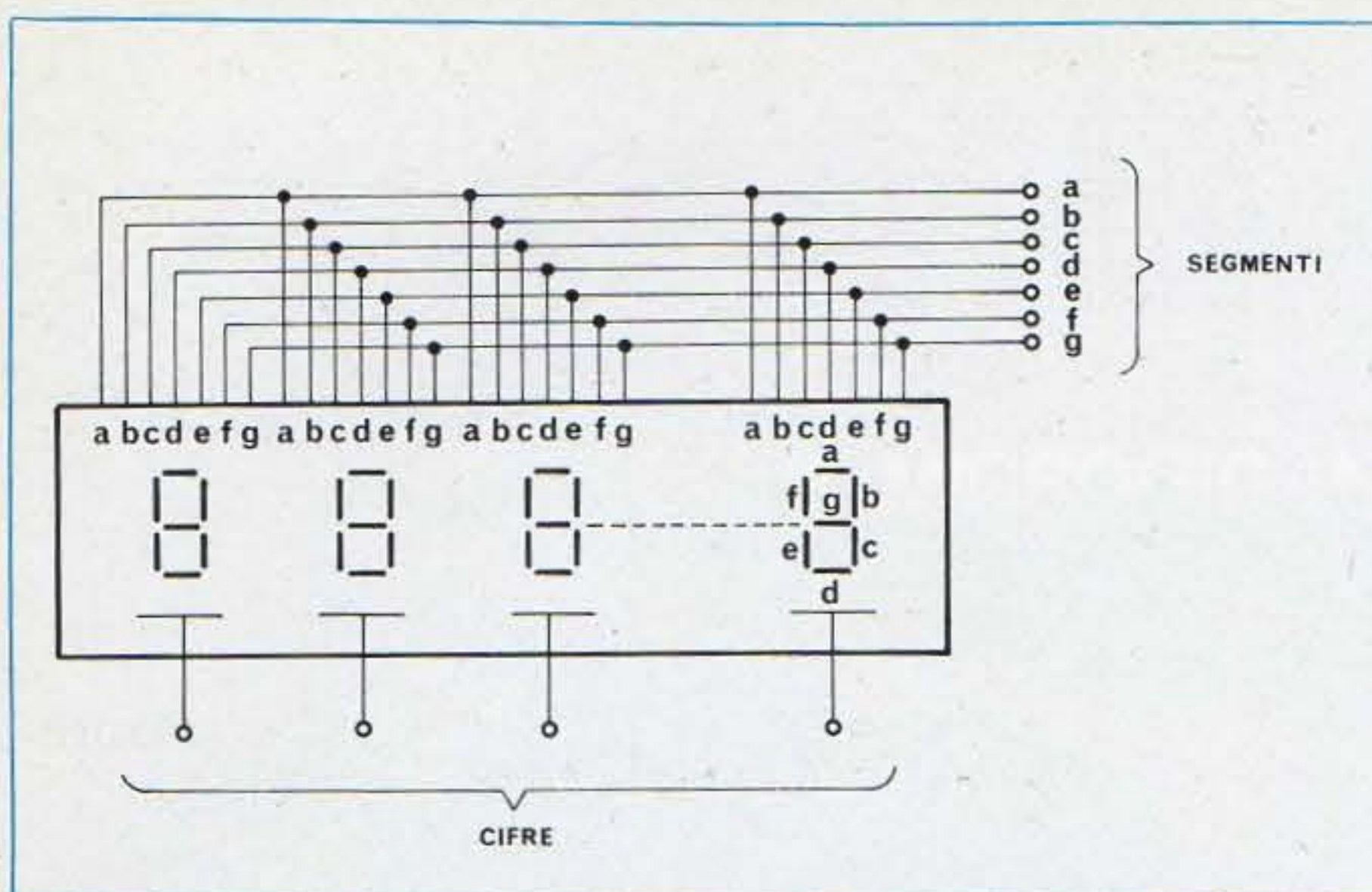



Figura 1 - Visualizzatore 7 - segmenti multiplato.

Questo articolo è il primo di una serie con la quale si intende far conoscere un'interessante esperienza didattico-applicativa realizzata nell'ambito dell'Istituto di Elettronica e Telecomunicazioni del Politecnico di Torino ed in collaborazione con il Gruppo Amatoriale MicroElaboratori, che attorno a tale esperienza si è costituito.

L'impostazione degli articoli è tale da individuare i criteri di base per il progetto hardware/software, soprattutto allo scopo di mettere chi intende realizzare "il picocomputer" nelle condizioni di capire ***quello che fa.***

LO scopo è anche quello di stimolare la formazione di gruppi di utenti o clubs o laboratori didattici nelle scuole a costi accettabili.

Importante: si ritengono già acquisite le conoscenze di base sul microprocessori.

- possibilità di introdurre dati ed inviare comandi,
- presentazione dei dati,

e occorrono per questo, al minimo, un visualizzatore ed una tastiera.

Una soluzione molto comoda e conveniente è modificare opportunamente una normale calcolatrice tascabile.
Di questa utilizzeremo appunto il visualizzatore, come organo di uscita, e la tastiera, come organo di ingresso. Molti microsistemi già esistenti utilizzano questo periferico che realizza un compromesso costo/prestazioni molto soddisfacente.

Il visualizzatore a 7 segmenti permette di presentare facilmente caratteri esadecimali, ed è quindi adatto per rappresentare con due cifre parole di 8 bit. Insieme ad una tastiera con 20÷30 tasti e con un opportuno programma di gestione si ottengono così con minima spesa tutte le funzioni di monitor: caricamento e lettura della memoria e di registri, avvio di programmi di utente, lettura e scrittura su cassette magnetiche, breakpoint, ecc

Con una tastiera estesa ed un po' di buona volontà nel leggere il visualizzatore a 7 segmenti, si possono utilizzare anche tutti i caratteri alfabetici e, per esempio, lavorare anche in BASIC o editare programmi sorgente da assemblare.

L'articolo descrive in dettaglio come un periferico di questo tipo può essere collegato ad un microprocessore, e come va organizzato il software di gestione. La soluzione calcolatrice è comoda ma non è l'unica; con gli stessi circuiti e lo stesso software si possono utilizzare visualizzatori e tastiere distinti di vario tipo. In particolare, per dispositivi destinati all'insegnamento è conveniente orientarsi verso display di maggiori dimensioni, onde permetterne la visione contemporanea da parte di più persone.

Descrizione a blocchi

Il nostro periferico comprende:

- visualizzatori,
- tastiera,
- circuiti di interfaccia.

Il visualizzatore consta di un certo numero di indicatori a 7 segmenti. Nel nostro caso sono da preferire indicatori a LED (luce rossa), che sono direttamente interfacciabili con logiche TTL. Quelli a scarica (verdi) richiedono tensioni di qualche decina di volt ed i cristalli liquidi hanno problemi meccanici di collegamento.
Dato che non conviene utilizzare una coppia di fili per ciascun segmento di ogni cifra, i catodi (display a catodo comune) sono collegati insieme cifra per cifra, e gli anodi di segmento corrispondenti sono collegati tra le varie cifre, come in fig. 1.

Per far comparire una cifra in una determinata posizione si devono attivare contemporaneamente la

Figura 2 - Comando di un visualizzatore multiplato a 4 cifre.

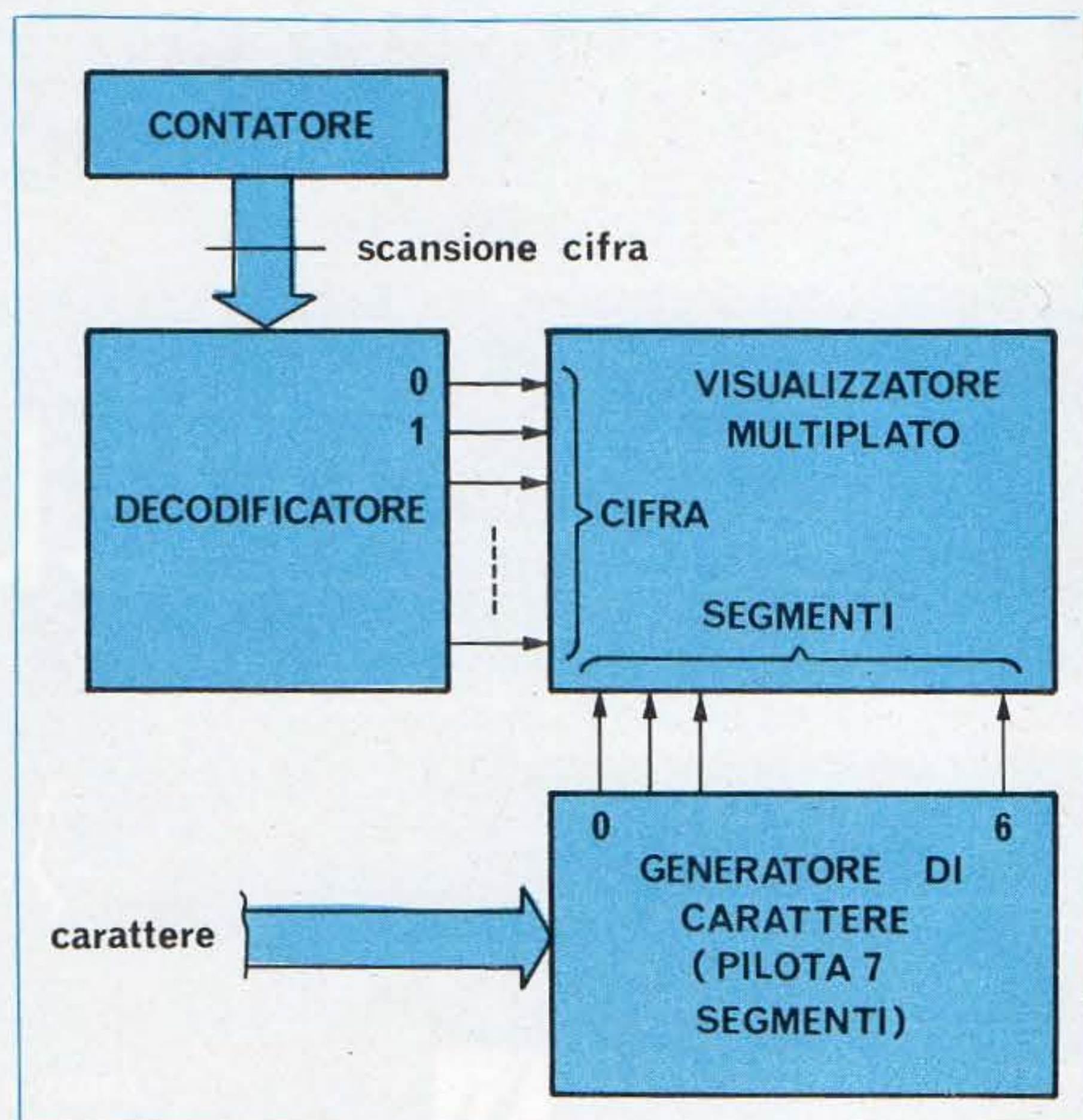



Figura 3 - Pilotaggio di un display multiplato.

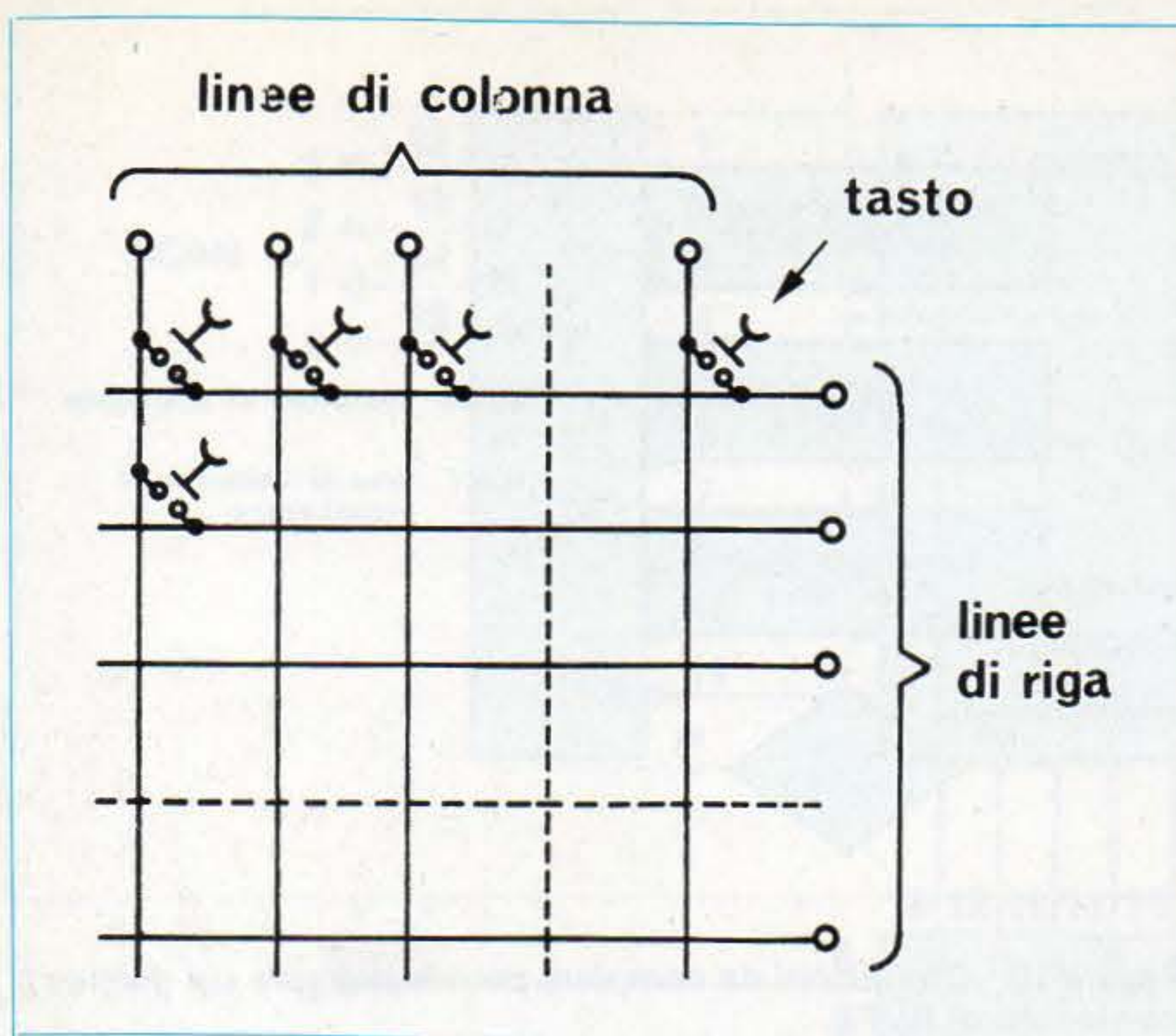



Figura 4 - Tastiera a matrice.

selezione di cifra ed i segmenti opportuni. Un numero completo di più cifre viene generato selezionando in successione le varie cifre ed inviando di volta in volta la codifica corrispondente sui segmenti (fig. 2).

Uno schema a blocchi che realizza queste funzioni è in fig. 3.

Noi vogliamo usare il visualizzatore per presentare dati elaborati dal microprocessore e quindi queste operazioni devono essere controllate da programma tramite le porte di I/O.

Anche per la tastiera non conviene usare un filo separato per ogni tasto, e quindi la configurazione più comune è quella a matrice (fig. 4). Premendo un tasto si connettono assieme una riga ed una colonna. Per generare il codice corrispondente a ciascun tasto possiamo impiegare lo schema funzionale della fig. 5

Supponiamo che le uscite del decodificatore e gli ingressi del codificatore siano attivi al livello basso (0), e che gli ingressi non collegati vadano al livello 1. Nella fig. 5 solo una delle uscite COD (cioé un filo di colonna) per volta va allo 0 logico.

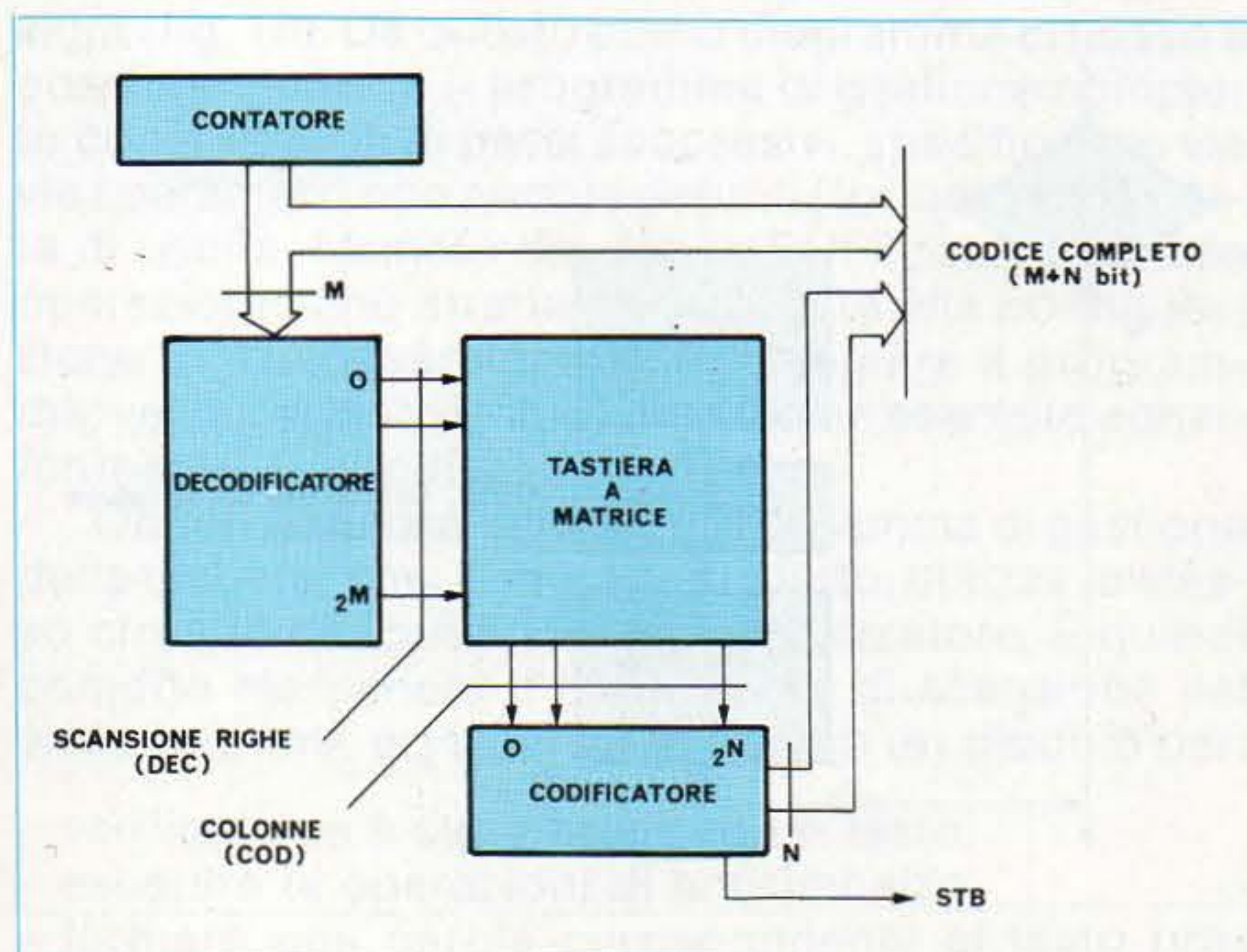



Figura 5 - Scansione e codifica di una tastiera a matrice.

Se ora viene premuto un tasto posto in questa colonna, lo 0 è riportato ad uno degli ingressi del codificatore che genera all'uscita una parola corrispondente al numero d'ordine dell'ingresso attivato ed un segnale STroBe, attivo ogni volta che uno qualsiasi degli ingressi va allo 0.STB può essere usato per indicare che un tasto è premuto (fig. 6). Il codice che specifica il tasto premuto si ottiene riunendo opportunamente l'uscita N del codificatore e l'ingresso M del decodificatore. Questo dato può essere memorizzato in un registro usando direttamente il segnale STB o, come nel nostro caso, letto dal microprocessore con una istruzione di INPUT.

Interfacciamento

Gli schemi di fig. 3 e 5 sono schemi funzionali e la separazione tra operazioni eseguite a programma o implementate ad hardware può essere fatta a vari livelli. Ad esempio, nel caso del display, possiamo far pilotare direttamente dal microprocessore la selezione di cifra ed i segmenti (ovviamente tramite adatti buffer per fornire le correnti richieste) (Fig. 7); in questo caso la scansione (contatore, decodifica) e la scelta dei segmenti da attivare per far comparire una determinata cifra (generatore di caratteri) sono realizzate a programma dal microprocessore. Un'altra

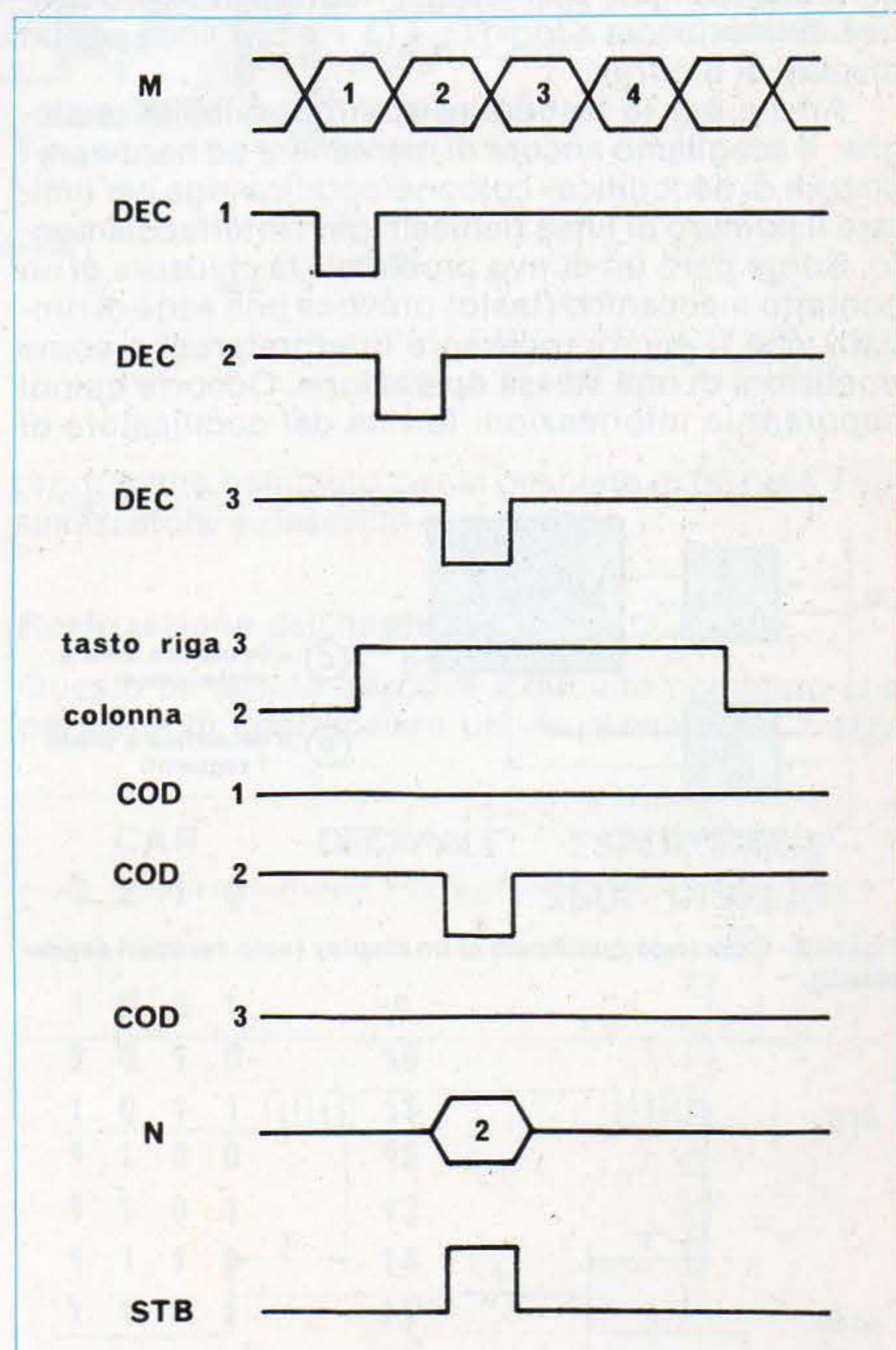



Figura 6 - Segnali nel circuito di Figura 4. DEC i e COD i sono attivi allo ∅.

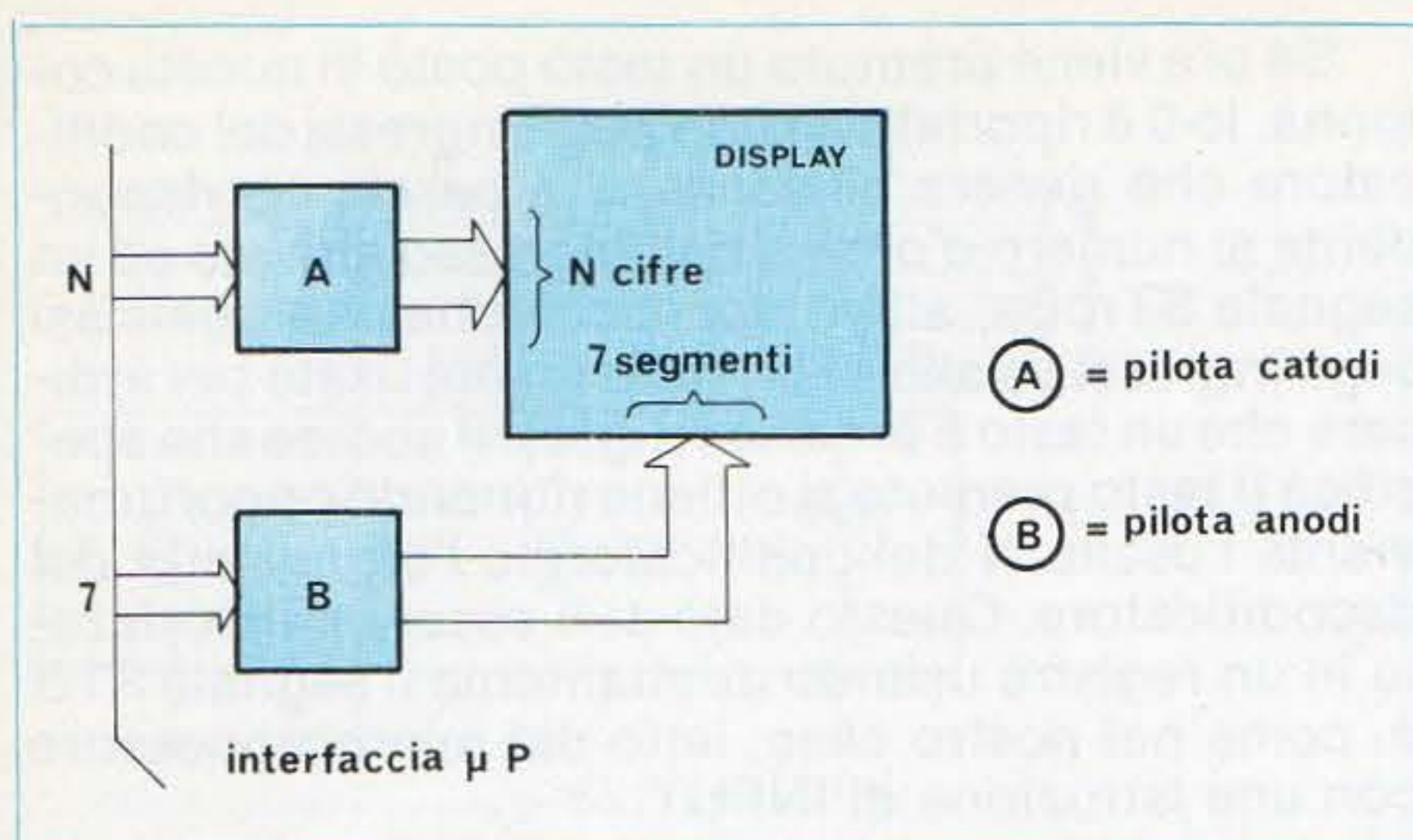



Figura 7 - Comando diretto di un display a catodo comune.

soluzione può essere inviare la selezione codificata ed il codice del carattere, realizzando da hardware decodifica della cifra e dei segmenti (Fig. 8). Ancora, potremmo usare un contatore esterno a cui inviare solo i comandi di reset e clock, o adottare soluzioni miste tra quelle indicate. Nel primo caso un display a 7 segmenti di N cifre occupa N + 7 linee di uscita e si ha il vantaggio di poter formare qualunque carattere o simbolo, perchè ogni segmento può essere selezionato individualmente da programma. La seconda soluzione è adottata nell'esempio del paragrafo seguente, perchè, limitandosi alla rappresentazione di cifre esadecimali, permette di ridurre il numero di linee di interfaccia a $\log_2 N + 4$ ($3 + 4 = 7$ linee per un display di 8 cifre).

Anche per la tastiera esistono possibilità analoghe, e scegliamo ancora di mantenere ad hardware i circuiti di decodifica- colonna/codifica-riga per limitare il numero di linee richiesto per l'interfacciamento. Sorge però un nuovo problema: la chiusura di un contatto meccanico (tasto) provoca una serie di rimbalzi che il microprocessore interpreterebbe come ripetizioni di una stessa operazione. Occorre quindi depurare le informazioni fornite dal codificatore di

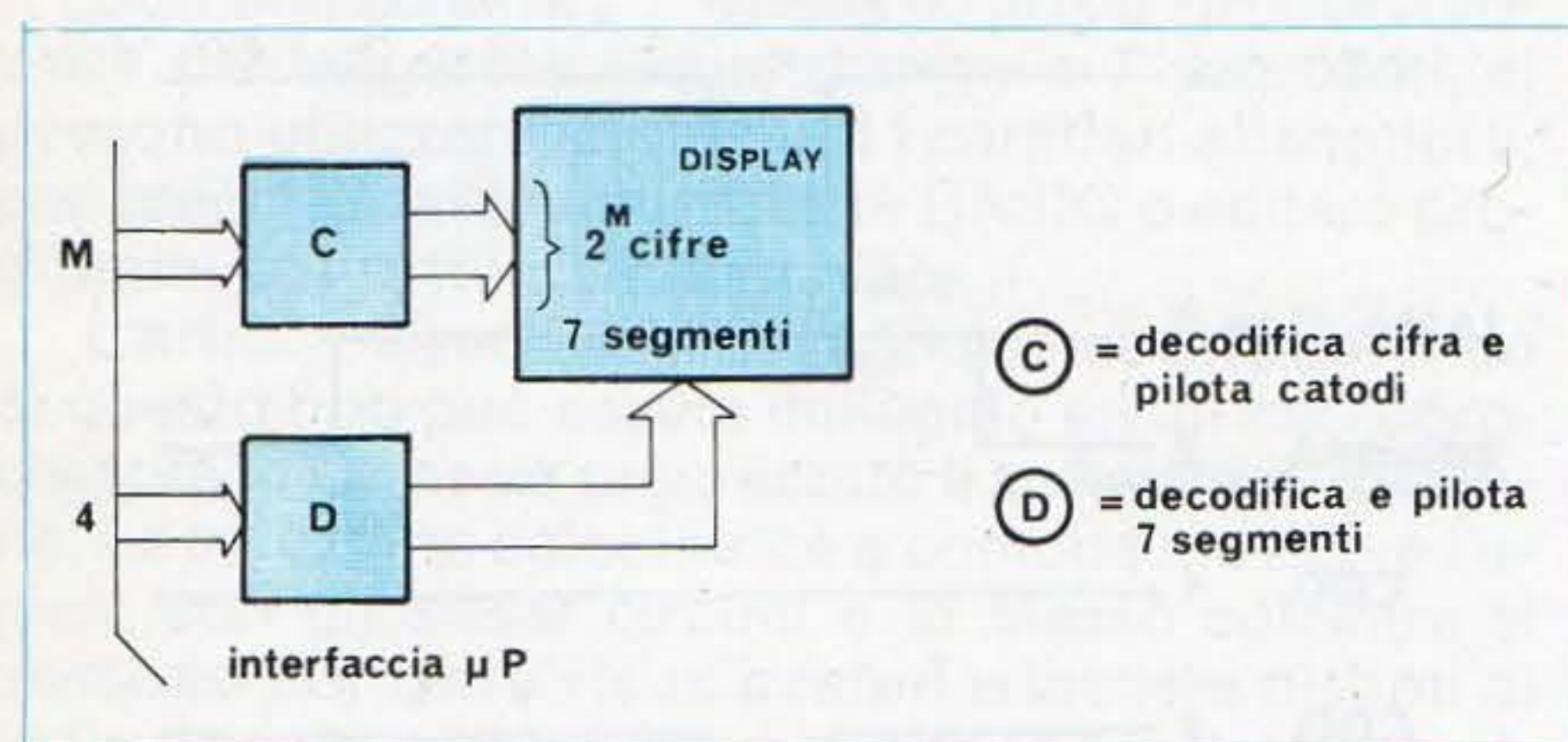



Figura 8 - Comando codificato di un display (solo caratteri esadecimali).

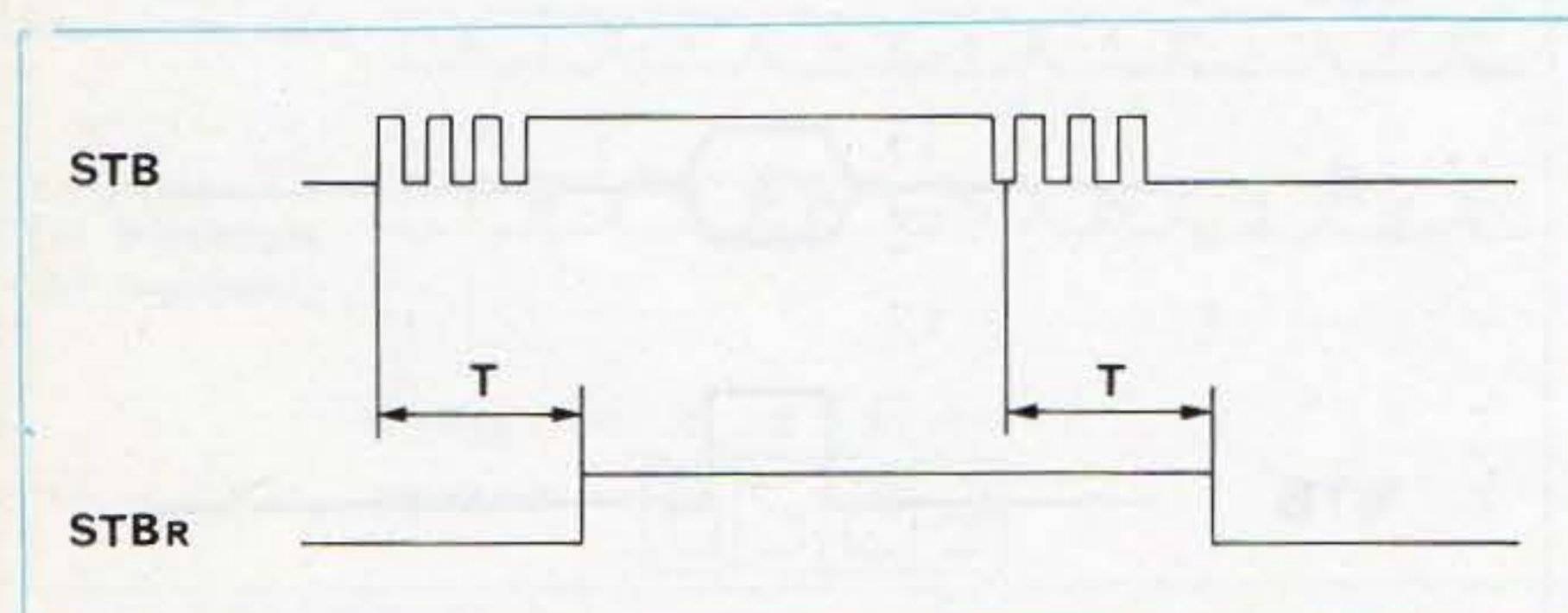



Figura 9 - Rimbalzi sul segnale STB e segnale ricampionato STBR.

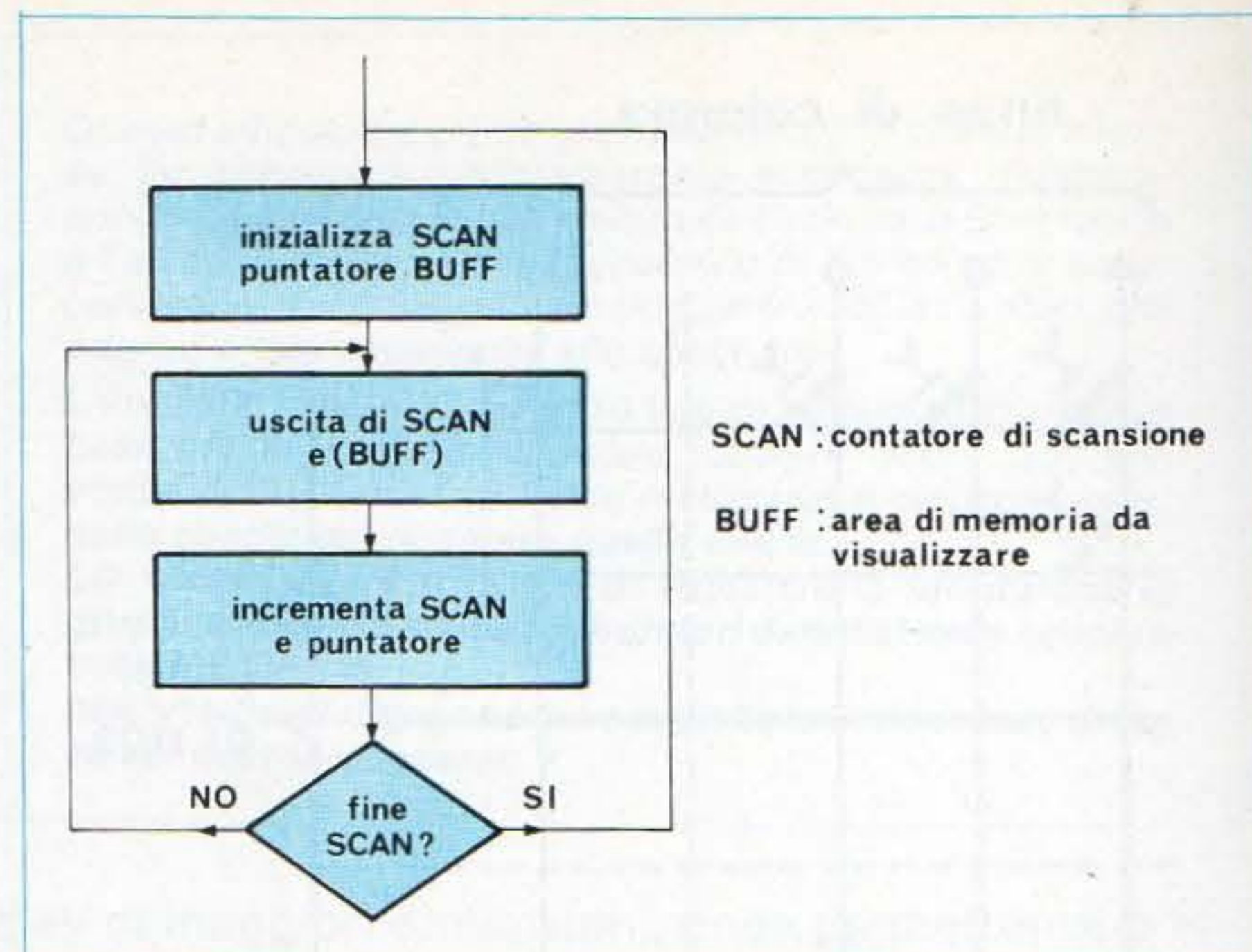



Figura 10 - Operazioni da compiere per visualizzare sul display il contenuto di BUFF.

colonna (in particolare il segnale di STB) dei rimbalzi, ed anche questa è una operazione che può essere eseguita da hardware o da programma.

La tecnica più semplice per eliminare l'effetto dei rimbalzi è ricampionare la linea dopo un ritardo assegnato, tale da esaurire il transitorio a partire dalla prima transizione (fig. 9).

Tastiera e visualizzatore richiedono entrambi una scansione, rispettivamente per indirizzare le cifre e le colonne. Nello schema proposto in seguito questa operazione è eseguita per entrambi dallo stesso circuito, riducendo così ancora le linee di interfaccia.

Programma di gestione

Per il visualizzatore le operazioni da compiere sono:

a) scansione delle cifre;
b) comando dei segmenti a seconda della informazione da visualizzare.

Dal momento che, come detto prima, decodifica della scansione e generatore di caratteri (esadecimali)

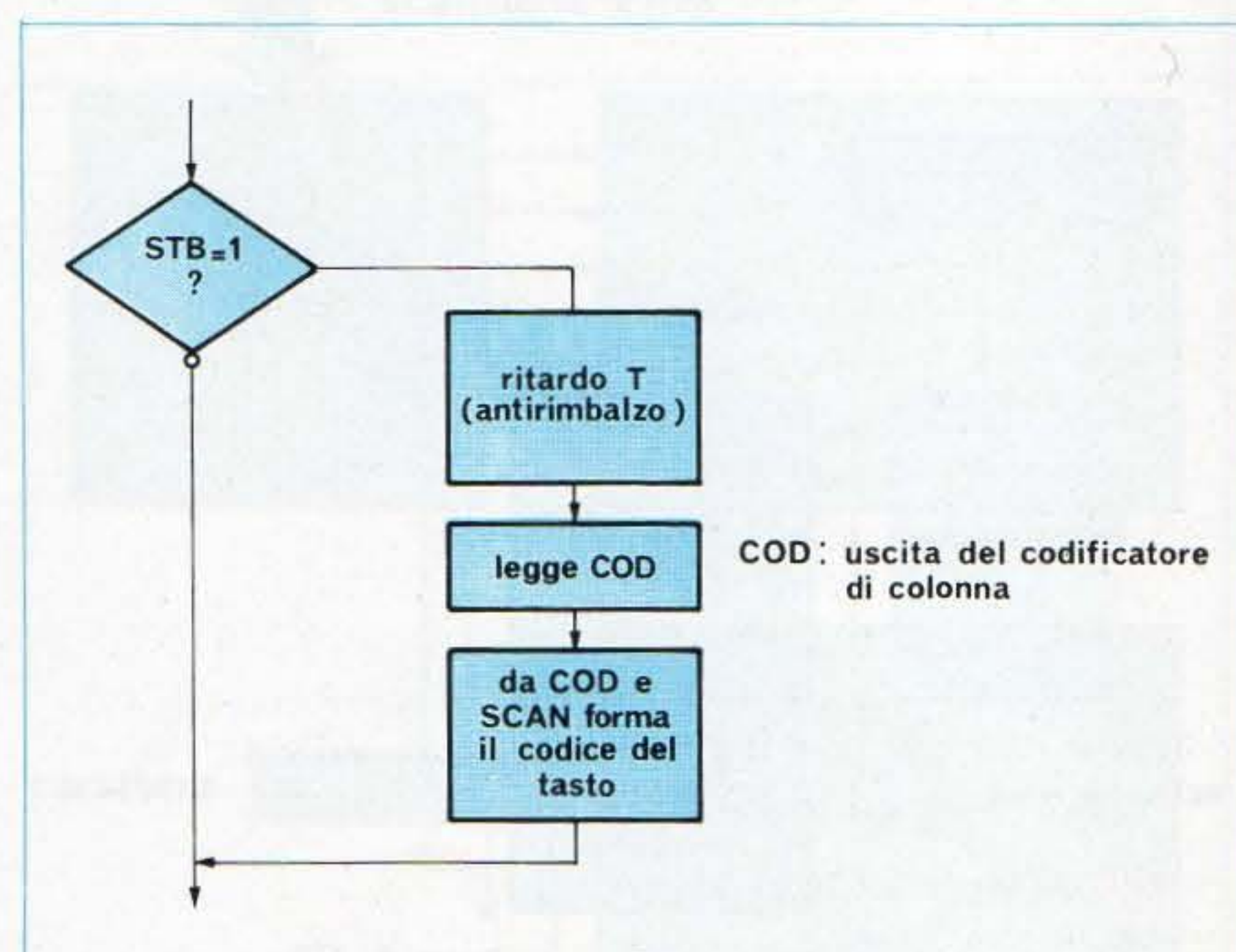



Figura 11 - Modulo software per la gestione della tastiera a matrice.

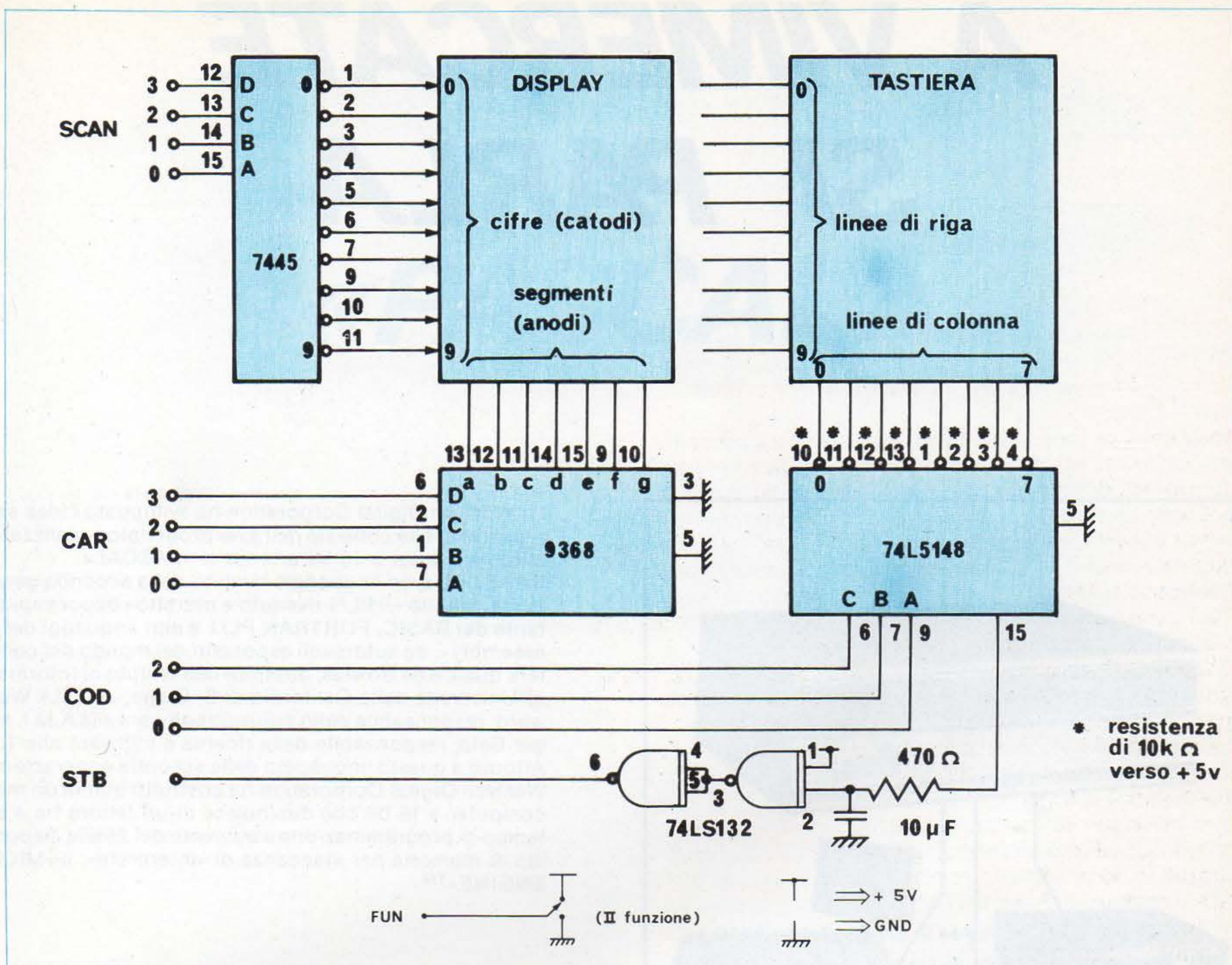



Figura 12 - Schema elettrico.

sono implementati da hardware, le funzioni da gestire da programma sono;

- il contatore di scansione (SCAN);
- i trasferimenti da un'area di memoria che contiene i dati da visualizzare (BUFF).

Possiamo ora tracciare la sequenza delle operazioni (fig. 10). Da questo primo diagramma di flusso è possibile ricavare il programma di gestione completo con una serie di passi successivi, specificando via via i parametri non ancora definiti (formato sulla porta di uscita, formato dei dati in BUFF, ecc.). Queste operazioni sono strettamente legate alla configurazione dell'hardware che dovrà eseguire il programma; verrà pertanto sviluppato solo un esempio con riferimento al circuito di fig. 12.

Occorre ancora definire il programma di gestione della tastiera, che, come è stato detto, utilizza lo stesso circuito di scansione del visualizzatore. È quindi comodo mantenere il programma di scansione del visualizzatore, ed inserire in questo un modulo per:

- verificare se è stato premuto un tasto,
- eseguire le operazioni di antirimbalzo,
- formare una parola corrispondente al tasto premuto.

Un primo schema a blocchi è nella figura 11; il programma completo per la gestione di tastiera e visualizzatore è descritto in seguito.

Realizzazione dell'hardware

Questo paragrafo descrive il circuito completo che permette di interfacciare un visualizzatore a 7 seg-

CAR 3 2 1 0	DECIMALE	ESADECIMALE (SUL DISPLAY)
1 0 0 1	9	9
1 0 1 0	10	A
1 0 1 1	11	b
1 1 0 0	12	C
1 1 0 1	13	d
1 1 1 0	14	E
1 1 1 1	15	F

Figura 13 - Rappresentazione dei caratteri esadecimali usando il decodificatore 9368.

CAR SCAN

3 2 1 0 3 2 1 0

7 0 PORTA DI USCITA

STB COD

2 1 0

7 0 PORTA DI INGRESSO

Figura 14 - Assegnazioni dei bit alle porte di I/0 per il programma di Figura 15.

menti, con catodo comune, con un massimo di 10 cifre, ed una tastiera fino a 80 tasti (organizzata su 10 righe di 8 colonne). Per lavorare con un minor numero di cifre o di tasti è sufficiente eliminare le connessioni non utilizzate; questo riduce anche il numero di linee di I/O richiesto per l'interfacciamento.

Lo schema elettrico completo del perifericocalcolatrice è in fig. 12. Le funzioni dei vari segnali sono:

- SCAN: scansione del visualizzatore e della tastiera (4 bit in codice BCD);
- CAR: carattere esadecimale da presentare sul visualizzatore (4 bit);
- COD: uscita del codificatore di colonna della tastiera (3 bit);
- STB: uscita del codificatore di tastiera indicante che è stato premuto un tasto.

I quattro bit SCAN vanno a pilotare i catodi del visualizzatore e le linee di riga della tastiera tramite un decodificatore BCD-decimale tipo 7445. Le uscite a collettore aperto di questo integrato sono in grado di assorbire 80 mA, garantendo così una buona luminasità ai segmenti. Il decodificatore-pilota dei segmenti (anodi) è un 9368 Fairchild. Questo è l'unico circuito che decodifica correttamente tutti i codici esadecimali (da 0000 a 1111); i caratteri da 9 a 15 sono rappresentati come in fig. 13.

Le colonne della tastiera sono collegate agli ingressi di un codificatore 8-3 tipo 74 LS148. Occorre inserire qui delle resistenze di pull-up (cioè collegate al + 5V) per garantire il livello di 1 logico alle linee aperte (tasti non premuti).

Le due porte inserite sulla linea STB inseriscono un piccolo ritardo, così da garantire che le uscite COD siano stabili quando STB è attivo.

Mediante un comando di seconda funzione possiamo raddoppiare il numero di codici impostabili dalla tastiera. Per questo utilizzeremo semplicemente un interruttore che pilota direttamente uno dei bit della porta di ingresso (FUN). Un altro comando permette di quadruplicare le funzioni e così via. Il meccanismo è analogo a quello dei tasti CONTROL e SHIFT nel codice ASCII; esistono i caratteri di base (32), e quelli ottenuti combinando con questi i bit di CONTROL e di SHIFT (128 caratteri in totale).

```
LOC   OBJ CODE M STMT SOURCE STATEMENT

                  1  ;
                  2  ;
                  3  ;***********************************************************************
                  4  ;
                  5  ;        VISTAST
                  6  ;        GESTIONE DI VISUALIZZATORE A 8 CIFRE
                  7  ;        E TASTIERA A MATRICE 6 RIGHE X 4 COLONNE
                  8  ;
                  9  ;
                 10  ;        PARAMETRI
                 11  ;
                 12  BUFF    EQU     0BF0H   ;MEMORIA VISUALIZZATA
                 13  MAXBUF  EQU     0F4H    ;LIMITE SUPERIORE DEL BUFFER
                 14  COMASK  EQU     6       ;MASCHERA DEL CONTATORE DI SCANSIONE
                 15  HIMASK  EQU     0F0H    ;MASCHERA 4 BIT ALTI
                 16  LOMASK  EQU     0FH     ;MASCHERA 4 BIT BASSI
                 17  SCMASK  EQU     70H     ;MASCHERA DELLA SCANSIONE
                 18  INMASK  EQU     03H     ;MASCHERA I BIT DI CODIFICA COLONNA
                 19  PORT    EQU     0FFH    ;INDIRIZZO DELLA PORTA DI IN-OUT
                 20  DELY    EQU     0A0H    ;RITARDO ANTIRIMBALZO
                 21  STB     EQU     7       ;POSIZIONE DI STB SU PORTA DI IN
                 22  ;
                 23  ;
                 24  ;        USO DEI REGISTRI
                 25  ;
                 26  ;        HL      PUNTATORE A BUFF
                 27  ;        L       CONTATORE SCAN PARI
                 28  ;        C,B     REGISTRI DATI TEMPORANEI
                 29  ;
                 30  ;
                 31  ;
                 32  ;        PROGRAMMA PRINCIPALE
                 33  ;        AD OGNI PASSO PORTA SUL VISUALIZZATORE
                 34  ;        SU DUE CIFRE, IL CONTENUTO DI UNA CELLA DI BUFF
                 35  ;
                 36  ;
0000  21F00B     37  MAIN    LD HL,BUFF      INIZIALIZZA IL PUNTATORE
0003  7D         38  LOOP    LD A,L
0004  FEF4       39          CP MAXBUF       ;FINE SCANSIONE ?
0006  28F8       40          JR Z MAIN       ;SI, RIPARTE DA INIZIALIZZAZIONE
0008  0F         41          RRCA
0009  0F         42          RRCA            ;SCALA DI TRE POSIZIONI A DESTRA
000A  0F         43          RRCA
000B  E606       44          AND COMASK      ;ISOLA IL CONTATORE SCAN PARI
000D  4F         45          LD C,A
000E  7E         46          LD A,(HL)       ;LEGGE IL CONTENUTO DI BUFF
000F  E60F       47          AND LOMASK      ;MASCHERA 4 BIT BASSI (CIFRA PARI)
0011  CD2300   R 48          CALL PER        ;FORMATTAMENTO E USCITA
0014  7E         49          LD A,(HL)       ;RILEGGE BUFF
0015  E6F0       50          AND HIMASK      ;MASCHERA I 4 BIT ALTI (CIFRA DISPARI)
0017  0F         51          RRCA
0018  0F         52          RRCA            ;SCALA DI QUATTRO POSIZIONI A DESTRA
0019  0F         53          RRCA            ;PER FORMARE CAR IN A
001A  0F         54          RRCA
001B  CBE7       55          SET 4,A         ;FORMA SCAN DISPARI
001D  CD2300   R 56          CALL PER        ;FORMATTAMENTO E USCITA
0020  2C         57          INC L           ;INCREMENTA PUNTATORE A BUFF
0021  18E0       58          JR LOOP
                 59  ;
                 60  ;
                 61  ;
                 62  ;***********************************************************************
                 63  ;
                 64  ;
                 65  ;        PER
                 66  ;        SOTTOPROGRAMMA DI FORMATTAMENTO
                 67  ;        VERIFICA SE E' PREMUTO UN TASTO
                 68  ;        FORMA IL CODICE CORRISPONDENTE
                 69  ;
                 70  ;
0023  81         71  PER     ADD A,C         ;FORMA IN A SCAN E CAR
0024  D3FF       72          OUT (PORT),A    ;USCITA CARATTERE
0026  E670       73          AND SCMASK      ;ISOLA SCAN PER FORMARE SUCCESSIVAMENTE IL
                 74                          ;CODICE DEL TASTO
0028  0F         75          RRCA            ;SCALA DUE POS. A SINISTRA
0029  0F         76          RRCA
002A  47         77          LD B,A          ;SALVA IN B
002B  CD3E00   R 78          CALL DELAY      ;RITARDO PER ANTIRIMBALZO
002E  DBFF       79          IN A,(PORT)     ;LEGGE IL PERIFERICO
0030  CB7F       80          BIT STB,A       ;TEST SU STB
0032  C8         81          RET Z           ;SE STB = 0 RITORNA
0033  57         82          LD D,A          ;SE STB = 1 SALVA IN D
0034  DBFF       83  TEST    IN A,(PORT)     ;RILEGGE E ATTENDE IL RILASCIO DEL TASTO
0036  CB7F       84          BIT STB,A
0038  20FA       85          JR NZ TEST
003A  7A         86          LD A,D
003B  E603       87          AND INMASK      ;MASCHERA I BIT DI CODIFICA COLONNA
003D  80         88          ADD A,B         ;FORMA IN A COD + SCAN
                 89  ;
                 90  ;
                 91  ;
                 92  ;        A QUESTO PUNTO A CONTIENE IL CODICE DEL TASTO PREMUTO
                 93  ;        NEL FORMATO:   7                   Ø
                 94  ;                      [  |  |  |  |  |  |  |  ]
                 95  ;
                 96  ;
                 97  ;
                 98  ;
                 99  ;                                 SCAN     COD
                100  ;
                101  ;        E' POSSIBILE INSERIRE L'ESECUZIONE DEI COMANDI
                102  ;
                103  ;
                104  ;        IL PROGRAMMA QUI INSERITO DEVE TERMINARE CON "RET"
                105  ;
                106  ;
                107  ;***********************************************************************
                108  ;
                109  ;
                110  ;
                111  ;        DELAY
                112  ;        RITARDO PER ANTIRIMBALZO
                113  ;
                114  ;
003E  C5        115  DELAY   PUSH BC
003F  06A0      116          LD B,DELY       ;CARICA IL PARAMETRO DI RITARDO
0041  10FE      117  HERE    DJNZ HERE       ;LOOP DI RITARDO
0043  C1        118          POP BC
0044  C9        119          RET
                120  ;
                121  ;
                122  ;
                123  ;        MODIFICHE PER USARE IL TASTO DI SECONDA FUNZIONE
                124  ;        FUN E@ SUL BIT 5
                125  ;
                126  ;
                127  INMASK  EQU     23      ;MASCHERA PER ACCETTARE ANCHE FUN
```

Figura 15 - Programma VISTAST.

Esempio di programma completo

Il listato di fig. 15 è un programma in assembler Z80 per la gestione di un periferico realizzato secondo lo schema di fig. 12 con:

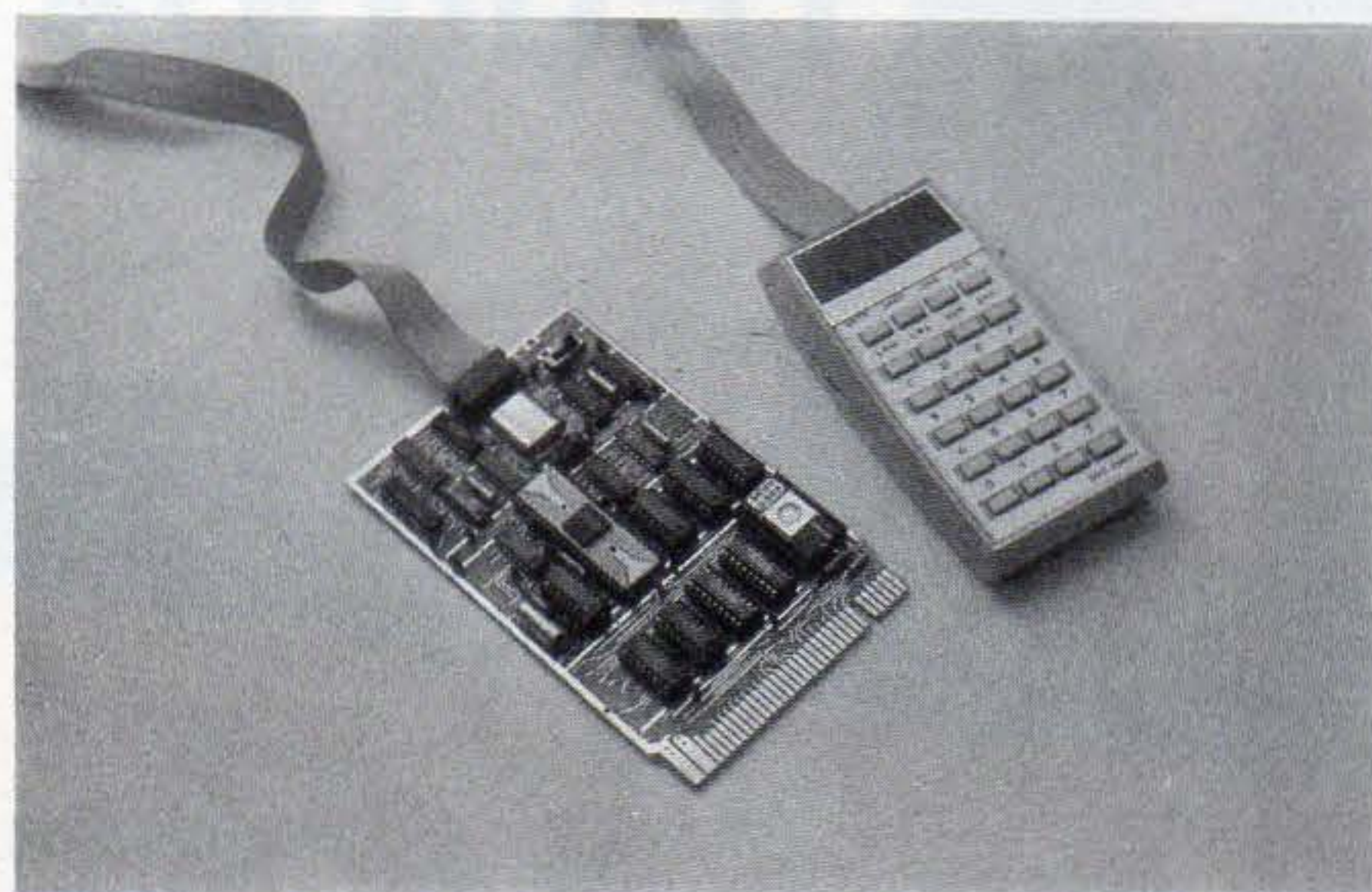
- visualizzatore ad 8 cifre,
- tastiera a 24 tasti.

Il display usa solo le prime 8 uscite del decodificatore 7445; la tastiera è una matrice di 6 righe per 4 co-

Il Picoperiferico

lonne. Il codificatore 74LS148 è usato solo per metà (4 ingressi), e l'uscita COD è quindi solo di due bit. Dato che la scansione è fatta solo su otto linee (perché il display ha 8 cifre e la tastiera 6 righe), l'ingresso D del decodificatore 7445 nello schema di fig. 12 va collegato a massa. Gli ingressi non utilizzati del codificatore 74LS148 vanno ad 1 attraverso le resistenze di pull-up. Calcolatrici da tasca con adatte caratteristiche (8 cifre, da 20 a 32 tasti), sono facilmente reperibili.

Il programma presenta continuativamente sul display, in esadecimale, il contenuto di quattro celle di memoria, da BUFF a BUFF + 3, usando due caratteri per il byte. Il contenuto di BUFF compare a partire dalle prime due cifre a destra (0 e 1).

Quando viene premuto un tasto il programma forma nell'accumulatore A il codice corrispondente al tasto specifico. È poi possibile usare il contenuto di A come dato di ingresso o come comando al programma. Una soluzione comoda per programmi di monitor è usare 16 tasti per introdurre dati in esadecimale e gli altri per inviare comandi.

Il Picosistema*

Il programma comprende un loop principale (MAIN) che provvede alla scansione del display ed a prelevare da BUFF i dati da visualizzare. Il sottoprogramma PER prepara i dati nel formato corretto e verifica se è stato premuto un tasto (linea STB); in quest'ultimo caso viene eseguito un modulo che identifica dati e comandi (non riportato), e da questo ritorna al programma principale. ■

* *La scheda di CPU è il mod. MBS 100 della Rad-Tronic-Torino.*

Nuove istruzioni dello Z-80

di **Franco Maddaleno**

Il set delle istruzioni dello Z-80 è più vasto di quello comunicato ufficialmente. Quali sono le nuove istruzioni e perché non sono state inserite? Acclusa una tabella che comprende tutti i codici dello Z-80.

Introduzione

Il microprocessore Z-80 ha la caratteristica di essere software compatibile con l'8080; nello stesso tempo il suo set di istruzioni è molto più ricco. La compatibilità è stata ottenuta mantenendo tutti i codici operativi dell'8080 e le nuove istruzioni sono state aggiunte nei "buchi"nella tabella dei codici lasciati dall'8080. Ad esempio il codice operativo 18 (esadecimale, come tutte le altre indicazioni che seguiranno) non ha alcun significato per l'8080, mentre fa eseguire un salto relativo allo Z80. Procedendo in questo modo si sarebbe potuto aggiungere allo Z80 solo una dozzina di istruzioni, tanti sono infatti i codici operativi non utilizzati dall'8080.

Per superare questa limitazione, alcuni di questi codici (e precisamente CB, DD, ED, FD) non sono stati considerati operazioni ma "prefissi": indicano cioè che il byte successivo è il vero e proprio codice, ed appartiene ad una classe particolare di istruzioni (per esempio il prefisso CB indica che il byte successivo è un'operazione di shift o su bit singolo).

Un ragazzo giovanissimo (15 anni), appartenente ad un club di microprocessori svizzero, aveva notato l'esistenza di altre istruzioni con prefisso nello Z80 e ne aveva parlato al Prof. Nicoud del Politecnico di Losanna, il quale lo aveva poi riferito ad un gruppo di appassionati di microprocessori di Torino, fra cui era il sottoscritto.

Nuove operazioni e nuovi registri

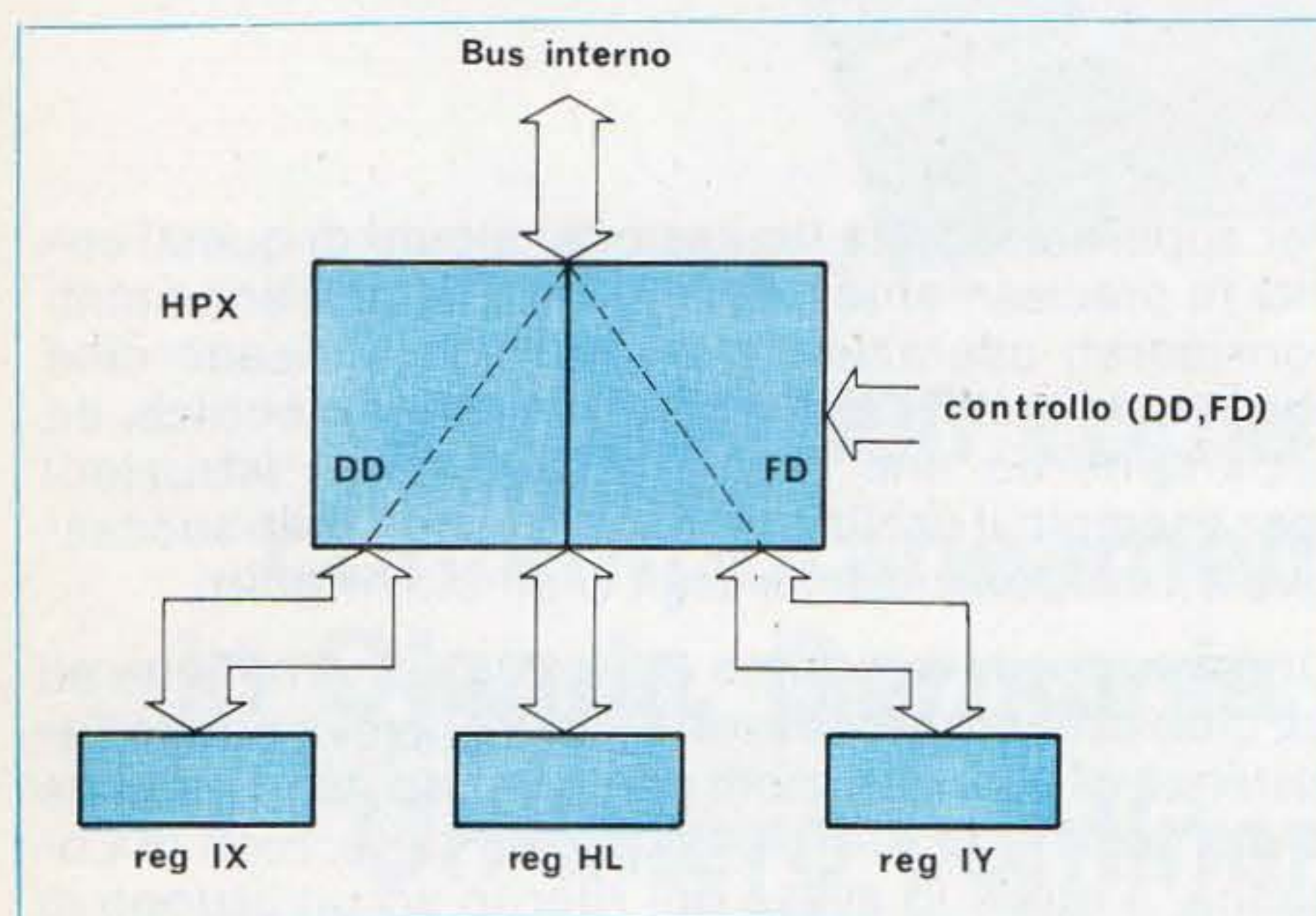
Iniziato lo studio sistematico dei codici operativi delle istruzioni con prefisso DD e FD, cioé delle operazioni che riguardano i registri indice, sono state notate le seguenti particolarità:

1) tutti i codici operativi che agiscono su IX cominciano col byte DD e quelli che agiscono su IY con FD.
2) Il byte successivo a DD o FD è sempre uguale al codice operativo della corrispondente operazione sulla coppia di registri HL (ad esempio PUSH HL ha codice operativo E5, mentre PUSH IX ha codice DD E5, e PUSH IY ha codice FD E5).
3) Quando il contenuto di IX o IY deve essere usato per indirizzare una cella di memoria, allora viene sommato al contenuto il terzo byte del codice operativo dell'istruzione (byte di offset) e il risultato è poi inviato sul bus degli indirizzi ottenendo così l'indirizzamento indicizzato.

Queste osservazioni hanno fatto nascere l'ipotesi che la struttura interna dello Z80 comprenda un multiplexer che, comandato dai prefissi DD e FD, collega al bus interno i registri indice IX o IY, mentre in assenza di prefissi collega la coppia HL (fig. 1). Inoltre, tutte le volte in cui il contenuto di un registro indice deve uscire sul bus degli indirizzi, viene attivato un sommatore che aggiunge al contenuto del registro lo offset letto dal terzo byte dell'istruzione e fa uscire il risultato.

Partendo dall'ipotesi che il multiplexer commuta semplicemente il collegamento interno dalla *coppia* HL al *registro* IX e IY, ci si è chiesto se fosse possibile considerare ognuno dei due registri indice (a 16 bit) a sua volta come una coppia di registri a 8 bit, proprio come HL può essere spezzato in H e L. Ovvero è possibile dividere ognuno dei registri indice in due registri da 8 bit *utilizzabili separatamente*, ottenendo così *quattro nuovi registri* a 8 bit?

Per verificare ciò si è fatta la seguente considerazione: prendiamo ad esempio l'operazione LD H, A (carica in H il contenuto di A: codice operativo 67): eseguendo questa operazione con il multiplexer commutato su IX e non su HL, il contenuto di A dovrebbe finire nella *parte alta* del registro IX.

Per commutare il multiplexer da HL a IX si è semplicemente preposto al codice operativo di LD H, A il prefisso DD: in definitiva si è visto che il codice operativo DD 67 provoca il caricamento della parte alta del registro IX con il contenuto dell'accumulatore.

Si è pensato di chiamare XH, XL, YH, YL le parti alte e basse dei registri IX e IY rispettivamente (fig. 2).

Nelle istruzioni di caricamento a 8 bit si sono trovati funzionanti tutti i trasferimenti tra XH, XL, YH, YL ed i registri A, B, C, D, E in entrambe le direzioni. Per ottenere questi trasferimenti basta preporrre al codice operativo della corrispondente istruzione su H o L il prefisso DD o FD.

Ad esempio LD A, YL ha come codice operativo FD 7D costruito in questo modo: FD è il prefisso che indica che ci si riferisce alla coppia YH-YL, 7D è il codice operativo di LD A, L. Analogamente LD XL, D ha codice DD 6A, dove DD è il prefisso che fa commutare il multiplexer dalla coppia HL alla coppia XH-XL e 6A è il codice operativo che, da solo, trasferisce il contenuto di D in L.

Bisogna osservare che le coppie HL, XH-XL, YH-YL sono mutuamente esclusive, in quanto il multiplexer ne collega una sola per volta: ciò significa che quando una istruzione di trasferimento posiziona il multiplexer su una coppia (ad es. XH-XL) le altre coppie indirizzabili dal multiplexer (in questo esempio HL e YH-YL) rimangono isolate e non raggiungibili dai dati: le istruzioni di trasferimento fra i registri HL, XH-XL, YH-YL non esistono, poichè il multiplexer può collegare una sola coppia per volta. Ci sono però i trasferimenti all'interno delle singole coppie, cioè da XH a XL, da YH a YL e viceversa, poiché il multiplexer attiva la coppia di registri XH-XL o YH-YL e l'istruzione funziona come LD H, L o LD L, H, agendo però sulle due metà di ognuno dei registri indice.

Sono state tentate combinazioni del tipo DD FD . . . per ottenere l'accesso contemporaneo alle coppie XH-XL e YH-YL, ma senza risultati: in effetti ciò è logico se si pensa al multiplexer che collega una sola coppia fra le tre possibili.

Sono pure stati cercati, ma senza esito, codici che dessero un caricamento dalla memoria ai singoli registri XH, XL, YH, YL e viceversa. Analogamente ai trasferimenti tra registri esistono anche i caricamenti con dato immediato dei singoli registri XH, XL, YH, YL ottenuti combinando i caricamenti immediati di H e L (codici operativi 26 e 2E) con i soliti prefissi DD e FD. Ad esempio LD YH, 03H ha codice FD 26 03.

Oltre alle operazioni di caricamento si è pure provato ad usare i "nuovi" registri XH, XL, YH, YL con le operazioni aritmetiche ad 8 bit ottenendo pieno successo. In altre parole, dal punto di vista delle operazioni aritmetiche e logiche ad 8 bit (ADD, ADC, SUB, SBC, AND, XOR, OR, CP, INC, DEC) questi nuovi registri si comportano esattamente come gli altri, dando gli stessi risultati e attivando gli stessi flag. Per impiegare i registri XH, XL, YH, YL si segue il solito metodo di utilizzare il codice operativo della corrispondente operazione su H o L preceduto dal prefisso DD o FD. Ad esempio ADD A, XL ha codice operativo DD 85, dove DD fa commutare il multiplexer sulla coppia

Fig. 1

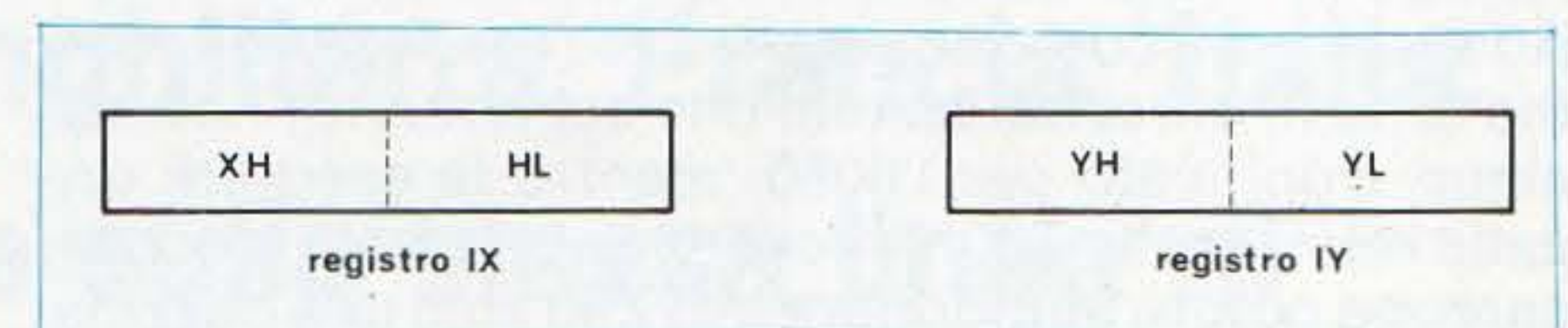



Fig. 2

XH-XL e 85 è il codice operativo di ADD A, L. Ancora si ha, ad esempio, che SBC A, XH ha codice DD 9C, XOR YH diventa FD AC e ancora INC YL diventa FD 2C. I registri XH XL (e YH YL) sono fra di loro separati come H e L: se ad esempio in XL è presente il valore FF e si fà l'operazione INC XL, il valore passa a 00, il contenuto di XH *non viene modificato* e i flag vengono posizionati come se la stessa situazione si fosse verificata su un altro registro. Si hanno dunque a disposizione 4 nuovi registri a 8 bit in più con cui si possono fare tutte le operazioni aritmetiche, logiche, i caricamenti immediati e quasi tutti i trasferimenti con gli altri registri.

Programma di verifica

Le prove sono state fatte su un sistema a scheda singola con monitor residente. Per controllare l'esatto funzionamento delle operazioni si è scritto un programma (vedi listato 1) che inizializza tutti i registri con operazioni di POP (in modo da inizializzare anche i flag), fa eseguire l'operazione in prova (che si trova in una subroutine), scambia il set principale di registri con quello alternativo ed esegue la stessa operazione su HL chiamando la subroutine a partire al secondo byte in modo da saltare il prefisso, scambia di nuovo i registri e li salva tutti con operazioni di PUSH e si ritorna a monitor, in modo da poter permettere l'esame dei risultati salvati in memoria.
La doppia operazione sui nuovi registri e su HL è necessaria per verificare che i flag siano posizionati dalle nuove operazioni esattamente come per le corrispondenti operazioni su HL: per questo bisogna inizializzare allo stesso modo HL, il registro indice su cui si lavora e i flag principali e altenativi. Questa tecnica permette pure di verificare che nessun registro venga accidentalmente sporcato dalla operazione in corso di prova.
Dopo l'operazione in prova è stata posta una istruzione LD BC, 7606H che ha codice operativo 01 06 76. In tal modo si può vedere se l'istruzione in prova richiede il byte di offset.
Se non viene richiesto si ritrova in BC il valore 7606, se invece l'operazione in prova richiede l'offset, il byte con 01 viene inglobato nell'istruzione precedente e l'istruzione successiva diventa 06 76 cioè LD B, 76H e il registro C non viene caricato. Se poi l'istruzione in prova richiede due byte in più dopo il codice operativo rispetto a quella su HL, si ha una situazione anomala e l'istruzione che rimane è 76 cioè HALT. Ovviamente al termine di ogni prova in BC' c'è sicuramente il valore 7606, poiché l'operazione di controllo su HL non richiede mai l'offset.

```
LOC   OBJ CODE M STMT SOURCE STATEMENT

                  1         ;
                  2         ;
                  3         ;   PROGRAMMA DI PROVA PER LE NUOVE ISTRUZIONI
                  4         ;   SUI REGISTRI XH,XL,YH,YL.
                  5         ;   PRIMA DI FARLO PARTIRE, BISOGNA CARICARE DALL'
                  6         ;   INDIRIZZO STK+1 I VALORI DI INIZIALIZZAZIONE
                  7         ;   DEI REGISTRI IN QUEST'ORDINE:
                  8         ;   I,FLAG ALTERNATIVI,A',C',B',E',D',L',H',
                  9         ;   IX,IY, (CIOE' XL,XH,YL,YH),FLAG PRINCIPALI,
                 10         ;   A,C,B,E,D,L,H.
                 11         ;
                 12         ;
0800             13         ORG     800H
0800  314F08     14         LD      SP,STK          ;INIZ. LO STACK POINTER
0803  F1         15         POP     AF              ;INIZ. IL REGISTRO I
0804  ED47       16         LD      I,A
0806  F1         17         POP     AF              ;INIZ. IL SET ALTERNATIVO
0807  C1         18         POP     BC              ;DI REGISTRI
0808  D1         19         POP     DE
0809  E1         20         POP     HL
080A  DDE1       21         POP     IX              ;INIZ. I REGISTRI
080C  FDE1       22         POP     IY              ;INDICE
080E  08         23         EX      AF,AF'
080F  D9         24         EXX
0810  F1         25         POP     AF              ;INIZ. I REGISTRI
0811  C1         26         POP     BC              ;PRINCIPALI
0812  D1         27         POP     DE
0813  E1         28         POP     HL
0814  CD2E08     29         CALL    TEST            ;FA ESEGUIRE L'OP. IN PROVA
0817  CD2F08     30         CALL    TEST1           ;CON E SENZA PREFISSO
081A  E5         31         PUSH    HL              ;SALVA I REGISTRI
081B  D5         32         PUSH    DE              ;PRINCIPALI
081C  C5         33         PUSH    BC
081D  F5         34         PUSH    AF
081E  FDE5       35         PUSH    IY              ;SALVA I REGISTRI
0820  DDE5       36         PUSH    IX              ;INDICE
0822  D9         37         EXX
0823  08         38         EX      AF,AF'
0824  E5         39         PUSH    HL              ;E QUELLI ALTERNATIVI
0825  D5         40         PUSH    DE
0826  C5         41         PUSH    BC
0827  F5         42         PUSH    AF
0828  ED57       43         LD      A,I
082A  F5         44         PUSH    AF
082B  C30000     45         JP      MONIT           ;TORNA A MONITOR
                 46         ;
                 47         ;
                 48         ;   LA SUBROUTINE CONTIENE L'ISTRUZIONE IN PROVA
                 49         ;   CHE VIENE FATTA ESEGUIRE SUI NUOVI REGISTRI,E
                 50         ;   PER CONTROLLO, ANCHE SU HL DEL SET ALTERNATIVO
                 51         ;   DI REGISTRI. E' STATA USATA LA PSEUDOISTRUZIONE
                 52         ;   DI 'DEFINE BYTE', POICHE' L'ASSEMBLATORE NON
                 53         ;   RICONOSCE LE NUOVE ISTRUZIONI E INUOVI REGISTRI.
                 54         ;
                 55         ;
082E  DD         56 TEST:   DEFB    0DDH            ;PREFISSO DD
082F  2C         57 TEST1:  INC     L               ;ISTR. IN PROVA: QUI INC XL
0830  010676     58         LD      BC,7606H        ;OFFSET (VEDI TESTO)
0833  D9         59         EXX
0834  08         60         EX      AF,AF'
0835  C9         61         RET
                 62 STK:    EQU     84FH            ;DEFINIZIONE DELLE VARIABILI
                 63 MONIT:  EQU     0H
                 64         END
%
```

Listing 1 - Programma di prova per le nuove istruzioni.

Altre prove

Considerando il risultato ottenuto, è stata osservata meglio la tabella dei codici operativi, notando in essa la presenza di numerosi "buchi".
Si è provato a tappare alcuni di questi ottenendo nuove istruzioni. Ad esempio le operazioni di *shift, rotate e manipolazione di bit* cominciano tutte col prefisso CB a cui segue un byte di codice operativo vero e proprio. Esistono, in questa classe di istruzioni, tutti i codici da 00 a FF eccetto quelli da 30 a 37 ("mancano" cioè i codici operativi da CB 30 a CB 37). Dando in pasto al microprocessore questi codici è stato trovato un nuovo tipo di shift che sposta a sinistra di un bit il contenuto di un registro o di una cella indirizzata dal contenuto di HL, mandando il bit più significativo nel flag di carry e facendo entrare un 1 a destra (fig. 3).
Questo shift si comporta al contrario di uno shift sinistro aritmetico o logico (che fa entrare a destra degli 0) facendo entrare degli 1: per questo è stato chiamato SLI *(Shift Left Illogical)*. Il primo byte della nuova operazione è sempre CB, il secondo byte va da 30 a 37 per effettuare lo SLI rispettivamente su B, C, D, E, H, L, (HL), A. Anche questo shift può utilizzare l'indirizzamento indicizzato: SLI (IX+of) ha codice operativo DD CB of 36 e analogamente SLI (IY+of) ha codice FD CB of 36.
Un ultimo gruppo di nuove istruzioni è stato trovato modificando i codici delle operazioni di *Shift, Rotate, Set bit, Reset bit* (N.B. non delle *Test bit*). Le precedenti operazioni indicizzate hanno il codice operativo formato premettendo al corrispondente codice con indirizzamento indiretto (ad es. SET 6, (HL)) il prefisso DD o FD e inserendo al terzo posto il byte di offset. Riprendendo l'esempio precedente, SET 6,

CY
reg
1

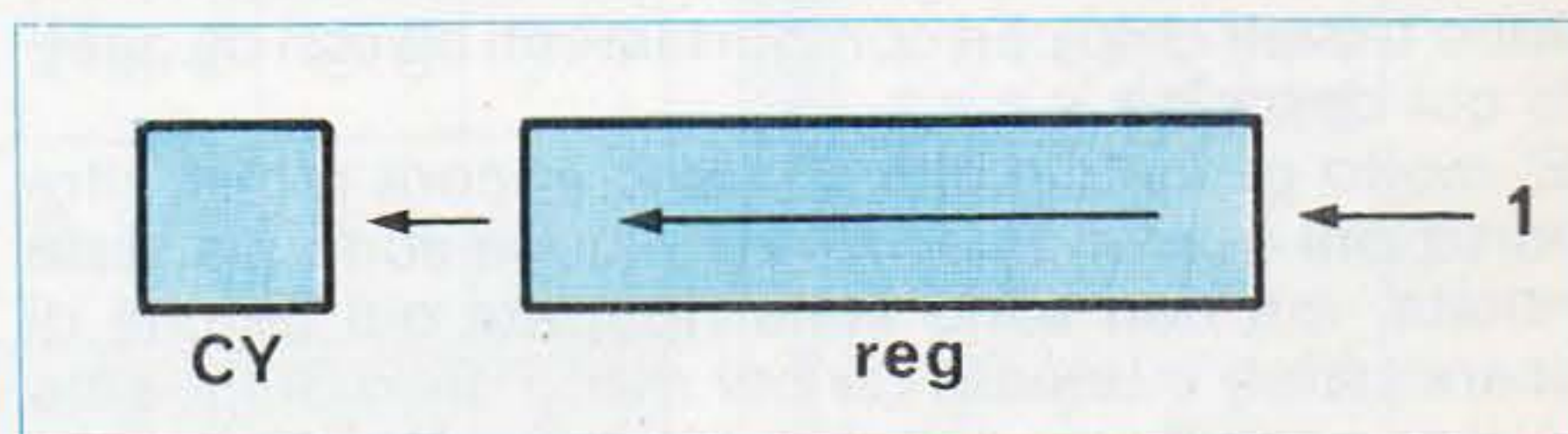



Fig. 3

(HL) con codice CB F6 diventa, indicizzata da IX, SET 6, (IX+of) con codice DD CB of F6. L'istruzione SET 6, D ha codice CB F2: non ha senso indicizzare una istruzione così poiché l'operando è interno alla CPU. Si è tuttavia provato a costruire una istruzione indicizzata con questo codice, ottenendo DD CB of F2: si è rilevato che quest'istruzione pone a 1 il sesto bit della cella indicizzata in memoria e contemporaneamente carica il risultato nel registro D. Equivale perciò all'istruzione SET 6, (IX+of) a cui segua LD D, (IX+of), occupando 4 byte al posto di 7 e impiegando 23 cicli di macchina invece di 42.
Tutte le istruzioni indicizzate del tipo *Shift, Rotate, Set bit, Reset bit* possono essere trasformate nelle corrispondenti istruzioni indicizzate *Shift and load, Rotate and load, Set bit and load* e *Reset bit and load* cambiando l'ultimo dei quattro bit del codice operativo, che nelle istruzioni originali termina sempre per 6 o per E: per aggiungere il caricamento in uno dei registri B, C, D, E, H, L bisogna modificare l'ultimo byte sottraendo rispettivamente 6, 5, 4, 3, 2, 1, mentre per ottenere il contemporaneo caricamento in A bisogna sommare 1 sempre all'ultimo byte.
Così ad esempio per eseguire SLI (IX+18H) e caricare contemporaneamente il risultato in E [ciò equivarrebbe a scrivere SLI (IX+18H) e LD E, (IX+18H)] si parte dal codice operativo di SLI (IX+18H) che è DD CB 18 36, e all'ultimo byte si sottrae 3 (per ottenere il caricamento in E) ottenendo DD CB 18 33. Analogamente si procede per tutte le altre istruzioni della serie, ad esclusione delle *Test bit* per le quali questo sistema non funziona poiché (probabilmente) la scrittura nel registro avviene contemporaneamente a quella in memoria nel sesto stato di macchina e le *Test bit* indicizzate hanno solo 5 stati di macchina.
Bisogna però osservare che queste operazioni (come ad esempio gli shift logici) non possono essere usate come primitive del tipo *Test and Set* per l'uso dei semafori via software nei sistemi multiprocessore, in quanto il processore può essere mandato in hold durante l'esecuzione della istruzione stessa.

Conclusione

Tutte queste istruzioni sono state cercate e provate su chip di produzione SGS e quindi funzionano sugli Z80 dell'SGS e della Zilog, poiché sono internamente identici.
Sono pure stati provati su chip di recente produzione Mostek e su uno dei primissimi chip prodotti dalla Zilog ottenendo sempre gli stessi risultati.
Bisogna però osservare che queste istruzioni "non ufficiali" dipendono da come è stato realizzato il circuito, cioè da come sono stati riempiti gli stati di *don't care* nelle varie tabelle che descrivono il funzionamento della macchina: ovviamente se vengono riprogettati i circuiti e se vengono cambiate le maschere, queste istruzioni possono cambiare o sparire completamente: bisogna però dire che finora non si sono trovati chip con comportamenti diversi da quello qui descritto.
E' molto probabile che ci siano ancora molte altre istruzioni nuove da scoprire: alcune sono già state notate, ma non sono state riportate qui perché di scarsissimo interesse (ancor meno dello SL I!), altre invece sono ancora completamente da scoprire: ad esempio le istruzioni con prefisso ED non sono ancora state esplorate.
Rimane un ultimo punto da considerare: perché queste istruzioni non sono state incluse nel set 'ufficiale' dello Z80? Un'ipotesi è che queste istruzioni, in particolarissime situazioni logiche, possano non funzionare: questo ne impedirebbe l'uso in campo professionale, mentre in campo hobbystico possono sempre essere utili. Questa è però solo una ipotesi che attende di essere verificata: finora non sono state trovate situazioni in cui queste istruzioni non funzionassero. Sarebbe interessante continuare queste prove per trovare nuove istruzioni ed eventuali limitazioni di funzionamento delle stesse, e gradirei conoscere i risultati ottenuti.
Rimango a disposizione per eventuali chiarimenti.

Tabella istruzioni Z80

La tabella comprende tutti i codici operativi eseguiti dal microprocessore Z 80, raccolti su un'unica pagina. Il compattamento sfrutta alcune caratteristiche dei codici; in particolare per le istruzioni che usano i registri IX e IY viene indicato una volta per tutte il prefisso da applicare alla corrispondente istruzione con i registri HL. Gruppi di istruzioni con lo stesso prefisso sono riuniti in uniche tabelle; il prefisso da usare è specificato a lato della tabella ed indicato con un asterisco (*) nel codice operativo. Le istruzioni di *shift, rotate, bit test, set e reset* hanno una struttura completamente modulare e sono raggruppate in una tabella molto compatta che però richiede l'esecuzione di una o due addizioni per ottenere il codice operativo completo. Anche le istruzioni di *LoaD* potrebbero essere riunite in poche caselle con la stessa tecnica, ma si è preferito tenerle distinte perchè di uso molto frequente.
La tabella non sostituisce il manuale con la spiegazione completa delle istruzioni, anzi, una certa familiarità con questo è necessaria per usarla con scioltezza. E' comunque un comodo ausilio di consultazione rapida che evita di dover sfogliare pagine e pagine durante il debug di un programma o per assemblare a mano poche istruzioni.
I colori indicano diverse classi di istruzioni:
Verde indica operazioni eseguite anche con i registri IX ed IY anteponendo i prefissi DD e FD rispettivamente. Queste istruzioni con i registri indice comprendono solo operazioni dirette sui registri.
Giallo indica operazioni eseguite con i registri IX ed IY in cui però il registro è usato come puntatore alla memoria. In questo caso occorre anche specificare lo scostamento d. Il formato completo è: <prefisso>, <cod. op. per HL (I byte)>, <scostamento>, [cod. op. (II byte)] in cui il campo [ ] è opzionale.
Arancione indica altre istruzioni che accettano il prefisso e lavorano sui registri IX ed IY ma non sono riportate dai manuali (vedi articolo).
I riquadri **Blu** indicano istruzioni che sono eseguite anche dall'8080.

I valori forniti come dato immediato sono indicati dai simboli:

n : parole di 8 bit (istr. di trasferimento e aritmetiche),
nn: parole di 16 bit (istr. di trasfermento e di salto),
d : scostamento (8 bit) per i registri indice,
e : scostamento (8 bit) per istruzioni di salto relativo.

ISTRUZIONI Z80© 1978 DDC

LD	B	C	D	E	H	L	(HL)	A	
B	40	41	42	43	44	45	46	47	06,n
C	48	49	4A	4B	4C	4D	4E	4F	0E,n
D	50	51	52	53	54	55	56	57	16,n
E	58	59	5A	5B	5C	5D	5E	5F	1E,n
H	60	61	62	63	64	65	66	67	26,n
L	68	69	6A	6B	6C	6D	6E	6F	2E,n
(HL)	70	71	72	73	74	75	-	77	36,n
A	78	79	7A	7B	7C	7D	7E	7F	3E,n
ADD	80	81	82	83	84	85	86	87	C6,n
ADC	88	89	8A	8B	8C	8D	8E	8F	CE,n
SUB	90	91	92	93	94	95	96	97	D6,n
SBC	98	99	9A	9B	9C	9D	9E	9F	DE,n
AND	A0	A1	A2	A3	A4	A5	A6	A7	E6,n
XOR	A8	A9	AA	AB	AC	AD	AE	AF	EE,n
OR	B0	B1	B2	B3	B4	B5	B6	B7	F6,n
CP	B8	B9	BA	BB	BC	BD	BE	BF	FE,n
INC	04	0C	14	1C	24	2C	34	3C	
DEC	05	0D	15	1D	25	2D	35	3D	
IN	*40	*48	*50	*58	*60	*68	-	*78	DB,n
OUT	*41	*49	*51	*59	*61	*69	-	*79	D3,n

cond.	JP add.	JR disp.	CALL add.	RET
—	C3,nn	18,e	CD,nn	C9
NZ	C2,nn	20,e	C4,nn	C0
Z	CA,nn	28,e	CC,nn	C8
NC	D2,nn	30,e	D4,nn	D0
C	DA,nn	38,e	DC,nn	D8
P.O.	E2,nn		E4,nn	E0
P.E.	EA,nn		EC,nn	E8
P	F2,nn		F4,nn	F0
M	FA,nn		FC,nn	F8

	RST
0	C7
1	CF
2	D7
3	DF
4	E7
5	EF
6	F7
7	FF

LD	→ A	← A
(BC)	0A	02
(DE)	1A	12
(HL)	7E	77
(nn)	3A,nn	32,nn
I	* 57	* 47
R	* 5F	* 4F

LD	← nn	← (nn)	→ (nn)	HL	PUSH	POP
B C	01,nn	ED,4B,nn	ED,43,nn		C5	C1
D E	11,nn	ED,5B,nn	ED,53,nn		D5	D1
H L	21,nn	24,nn	22,nn		E5	E1
S P	31,nn	ED,7B,nn	ED,73,nn	F9		
A F					F5	F1

	BC	DE	HL	SP
ADD	09	19	29	39
ADC	* 4A	* 5A	* 6A	* 7A
SBC	* 42	* 52	* 62	* 72
INC	03	13	23	33
DEC	0B	1B	2B	3B

EX	AF	rp	HL
	08	D9	EB	DE
			E3	(SP)

IX → DD.
IY → FD.

* → ED

	I	IR	D	DR
LD	* A0	* B0	* A8	* B8
CP	* A1	* B1	* A9	* B9
IN	* A2	* B2	* AA	* BA
OUT	* A3	* B3	* AB	* BB

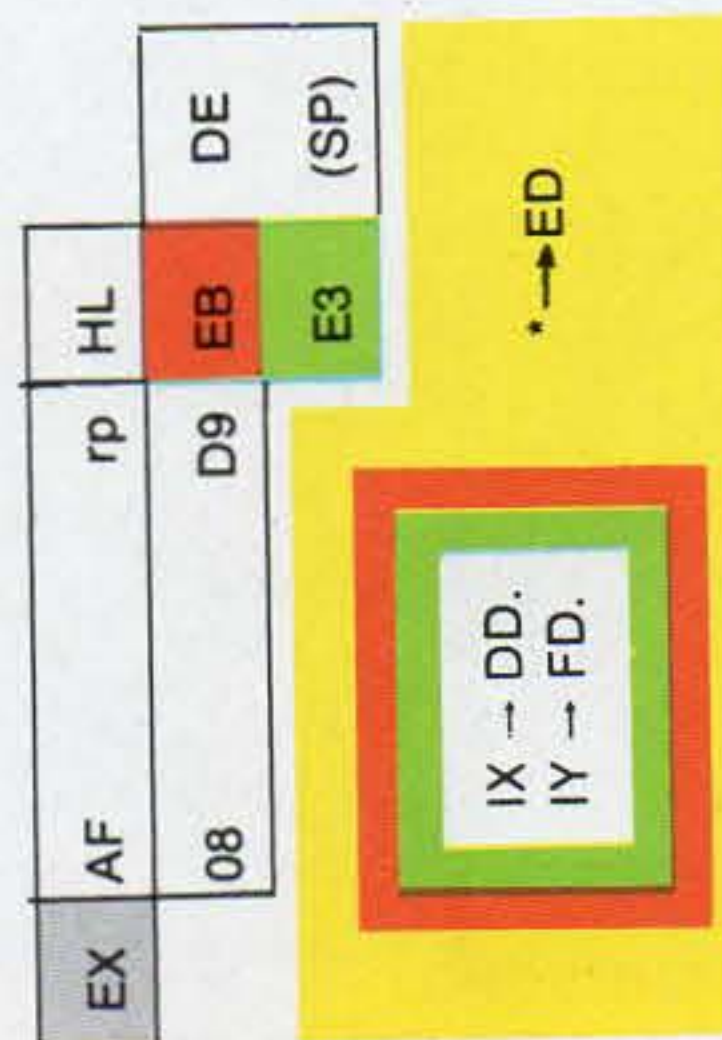
(DE) ← (HL)
A ↔ (HL)
(HL) ←
← (HL)

	A		cy Reg.
RLC	07	· 00	CB, Reg
RRC	0F	· 08	CB, Reg
RL	17	· 10	CB, Reg
RR	1F	· 18	CB, Reg
SCA		· 20	
SRA		· 28	
SLI		· 30	
SRL		· 38	
BIT		· 40	CB, · ,Reg, bit
RES		· 80	CB, · ,Reg, bit
SET		· C0	CB, · ,Reg, bit

DJNZ	10,e
JP (HL)	E9

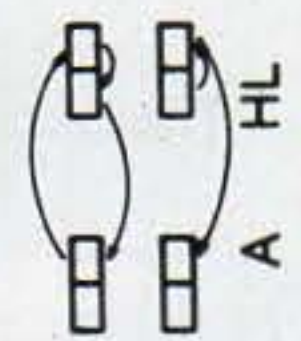
IM 0	* 46
IM 1	* 56
IM 2	* 5E

RETI	* 4D
RETN	* 45

DAA	27
CPL	2F
CCF	3F
SCF	37
NOP	00
HLT	76
DI	F3
EI	FB

Reg.	B	C	D	E	H	L	(HL)	A	
	0	1	2	3	4	5	6	7	bit
	00	08	10	18	20	28	30	38	

RLD	* 6F
RRD	* 67

A HL

NEG	* 44

Esempi di uso

Per indicare la tabella via via utilizzata si fa riferimento alla fig. 1.

A1 A2 A3
B1 B2 E
D A4 F
C1 C2 C3 G
H

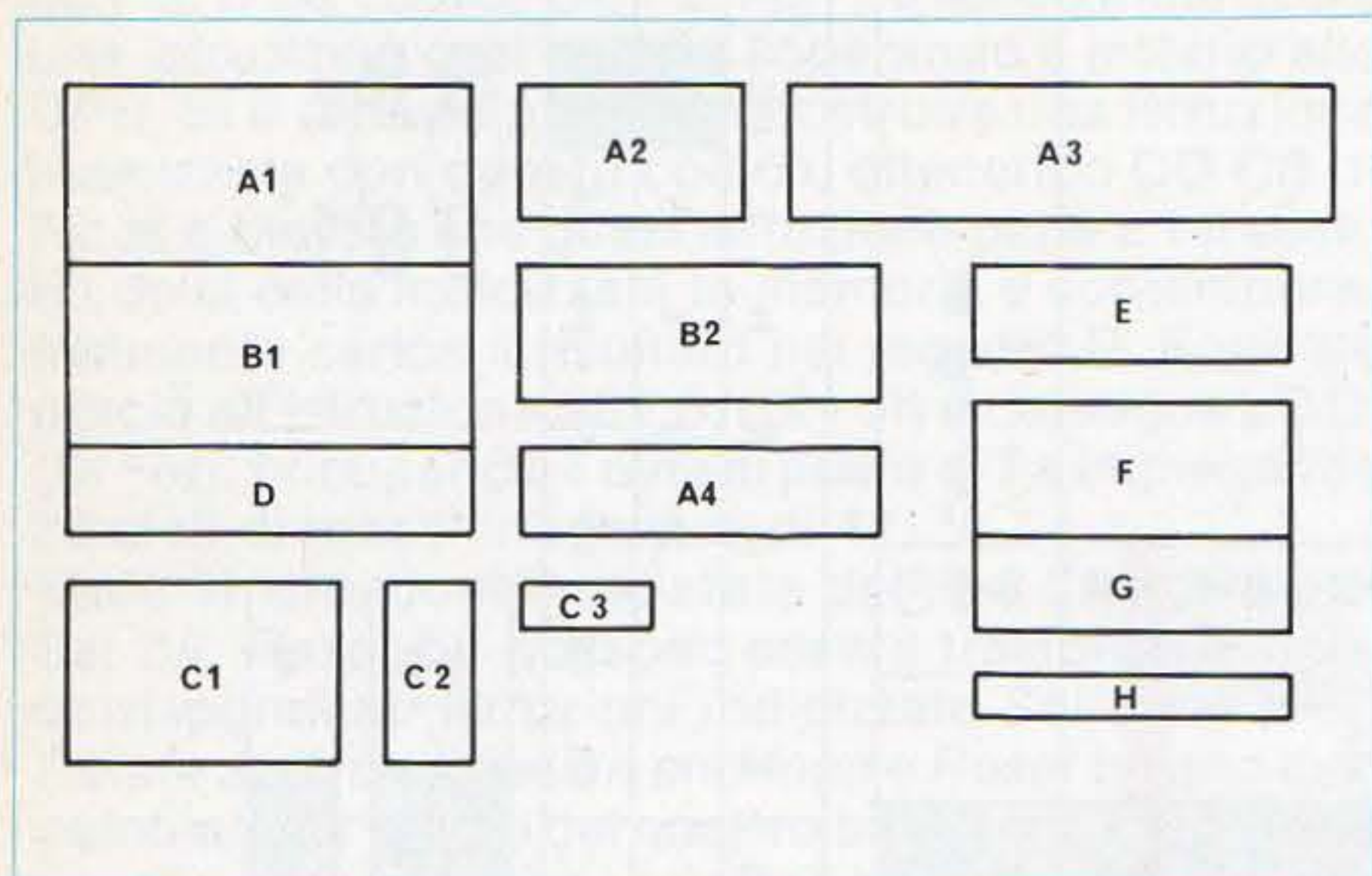



Figura 1 - Mappa istruzioni Z80.

Istruzioni di trasferimento

Tabella A1: trasferimenti tra registri, immediati e indiretti.
La sorgente è specificata sull'intestazione delle colonne, la destinazione sulle righe.
Esempio:

```
LD A, C        =     79
   ↑  ↑              colonna
   └─────────────── riga
LD E, n        =     1E, n
LD B, (IY+d)   =   FD,46 d
                   ↑  ↑  └─scostamento
                   │  └───codice per LD B, (HL)
                   └──────prefisso IY
```

Tabella A2: trasferimenti con l'accumulatore A.
La freccia indica il verso del trasferimento
→A: A è destinazione
←A: A è sorgente
Esempio: LD (BC), A = 02
LD A, (nn) = 3A, nn

Tabella A3: trasferimenti su coppie di registri, PUSH e POP.
La freccia indica il verso del trasferimento, analogamente alla tabella A2.
Esempi: LD (nn), DE = ED,53, nn
LD IX, nn = DD,21 nn
PUSH HL = E5

Tabella A4: istruzioni di scambio.
Esempi: EX AF, AF' = 08
EXX = 09
EX IY, (SP) = FD, E3

Istruzione aritmetiche e logiche

Tabella B1: operazioni su 8 bit.

Le righe specificano il codice mnemonico della operazione, le colonne l'operando.
Esempi: ADD A, (HL) = 86
CP n = FE, n

Tabella B2: operazioni su 16 bit.
Righe e colonne come per la tabella B1; i codici operativi preceduti da un punto richiedono il prefisso ED.
Esempi: ADD HL, DE = 19
SBC HL, BC = ED,42
ADD IX, BC = DD,09

Istruzioni di salto

Tabella C1: salti e chiamate di sottoprogrammi.
Le colonne specificano la condizione di salto, le righe il tipo di istruzione.
Esempi: JP 3200H = C3,00,32
CALL NZ 8231H = C4,31,82
JR NC +10 = 30,0A

Tabella C2: istruzioni di restart.
Tabella C3: altre istruzioni di salto.

Istruzioni di trasferimento sui periferici

Tabella D
L'uso è analogo alle istruzioni aritmetiche. Ricordare che, per le istruzioni di I/O con prefisso, l'indirizzo del periferico è nel registro C.
Esempi: IN A, (F3H) = DB,F3
OUT (C), D = ED, 51

Istruzioni su blocchi di dati

Tabella E: LoaD, ComPare, INput, OUTput.
Tutte queste istruzioni hanno il prefisso ED. Le indicazioni a lato richiamano le funzioni svolte.
Esempi: LDIR = ED,B0
INR = ED,AA

Istruzioni di rotazione e scalamento

Tabella F
Le istruzioni presenti nella prima colonna operano sull'accumulatore A. Quelle indicate nella seconda hanno prefisso CB, operano su tutti i registri ed il codice operativo completo si ottiene sommando (in esadecimale) la specificazione del registro (tabella H). È stata indicata anche l'istruzione SLI descritta nell'articolo.

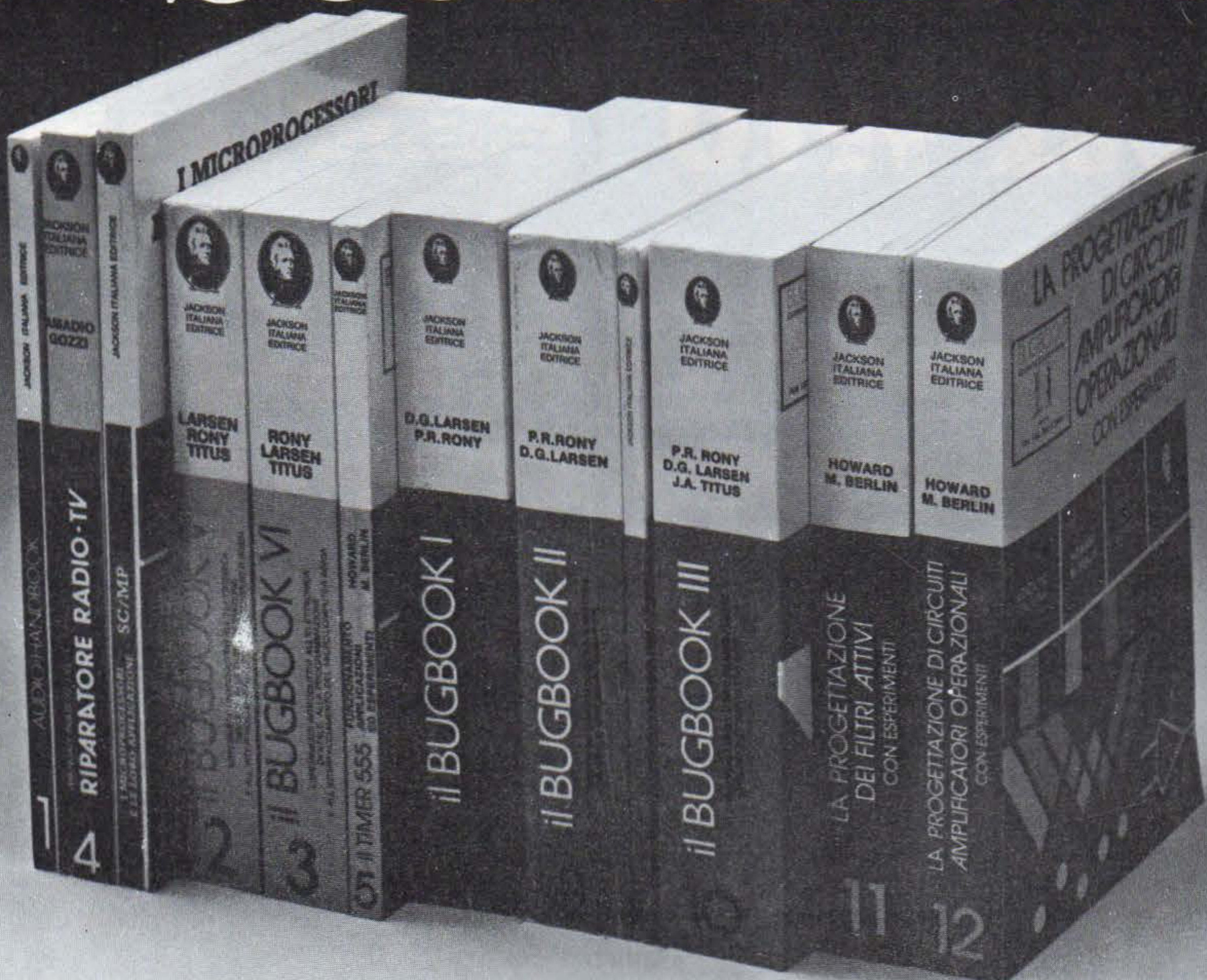
```
                              ┌──Cod. SCA
Esempi: RRCA   = OF           ↓   ↓──REG. C
SCA,C          = CB,21   21 = 20 + 1

RR, (IX+d)    = DD,CB, 1E   1E  18+6
                ↑
                └──────────────── prefisso IX
```

Istruzioni di manipolazione dei bit

Tabella G: BIT, RESet, SET
Anche queste istruzioni hanno prefisso CB ed il codice si ottiene sommando identificazione del registro e del bit (tabella H).

```
Esempi: BIT 6,A = CB,77   77 = 40 + 7 + 30
                                    ↑    ↑──bit 6
RES 0,H = CB,84                     └──────Reg. A
SET 5,(IX+d) = DD,CB d,EE   EE = CO + 6 + 28
               ↑                      ↑  bit 5
               │                        Reg. HL
               └──────────────────── prefisso IX
```

Le altre istruzioni sono indicate singolarmente essendo di uso diretto.

i "best-sellers"

AUDIO HANDBOOK

Un manuale di progettazione audio con discussioni particolareggiate e progetti completi.

L. 9.500 (Abb. L. 8.550)

MANUALE PRATICO DEL RIPARATORE RADIO-TV

Un autentico strumento di lavoro. Fra i numerosi argomenti trattati figurano: il laboratorio. Il servizio a domicilio. Antenne singole e centralizzate. Riparazione dei TV b/n e colore. Il ricevitore AM FM. Apparecchi e BF e CB. Strumentazione. Elenco ditte di radiotecnica, ecc.

L. 18.500 (Abb. L. 16.650)

SC/MP

Questo testo sul microprocessore SC/MP è corredato da una serie di esempi di applicazione e di programmi di utilità generale, tali da permettere al lettore una immediata verifica dei concetti teorici esposti e un'immediata sperimentazione anche a livello di realizzazione progettuale.

L. 9.500 (Abb. L. 8.550)

IL BUGBOOK V E IL BUGBOOK VI

Esperimenti introduttivi all'elettronica digitale, alla programmazione ed all'interfacciamento del microprocessore 8080A. I Bugbook V e VI costituiscono i primi veri testi organici a livello universitario sui microprocessori, con taglio nettamente sperimentale. Questi testi, oltre al Virginia Polytechnic Institute, sono utilizzati in corsi aziendali, in seminari di aggiornamento tecnico e in scuole di tutto il mondo.

L. 19.000 ogni volume (Abb. L. 17.100)

IL TIMER 555

Il 555 è un temporizzatore dai mille usi. Il libro descrive circa 100 circuiti utilizzanti questo dispositivo e numerosi esperimenti.

L. 8.600 (Abb. L. 7.750)

IL BUGBOOK I E IL BUGBOOK II

Strumenti di studio per i neofiti e di aggiornamento professionale per chi già vive l'elettronica "tradizionale", questi due libri complementari presentano esperimenti sui circuiti logici e di memoria, utilizzanti circuiti integrati TTL. La teoria è subito collegata alla sperimentazione pratica, secondo il principio per cui si può veramente imparare solo quello che si sperimenta in prima persona.

L. 18.000 ogni volume (Abb. L. 16.200)

IL BUGBOOK II/A

Esperimenti di interfacciamento e trasmissione dati utilizzanti il ricevitore/trasmettitore universale asincrono (Uart) ed il Loop di corrente a 20 mA.

L. 4.500 (Abb. L. 4.050)

IL BUGBOOK III

Questo libro fornisce una parola definitiva sull'argomento "8080A" divenuto ormai un classico nella letteratura tecnica sui microprocessori. Da ogni parte, sia da istituti di formazione che da varie case costruttrici, sono stati pubblicati manuali e libri di testo, ma nessuno raggiunge la completezza di questo Bugbook e, soprattutto, nessuno presenta l'oggetto "8080A" in un modo così didattico e sperimentale.

L. 19.000 (Abb. L. 17.100)

LA PROGETTAZIONE DEI FILTRI ATTIVI CON ESPERIMENTI

Tratta un argomento di notevole attualità, rendendolo piano e comprensibile a tutti. Le riviste di settore dedicano ampio spazio a questo aspetto dell'elettronica da oltre tre anni. Questo libro raccoglie tutto quanto è necessario sapere sui filtri attivi aggiungendovi numerosi esempi pratici ed esperimenti.

L. 15.000 (Abb. L. 13.500)

LA PROGETTAZIONE DEGLI AMPLIFICATORI OPERAZIONALI CON ESPERIMENTI

Gli amplificatori operazionali, in gergo chiamati OP-AMP, sono ormai diffusissimi in elettronica. Il libro ne spiega il funzionamento illustra alcune applicazioni pratiche e fornisce numerosi esperimenti. Le persone interessate all'argomento sono moltissime: dal tecnico esperto al semplice hobbista. Si tratta del miglior libro pubblicato nella materia specifica.

L. 15.000 (Abb. L. 13.500)

Bit

CEDOLA DI COMMISSIONE LIBRARIA Da inviare a Jackson Italiana Editrice srl - Piazzale Massari, 22 - 20125 Milano.

SCONTO 10% AGLI ABBONATI

☐ Inviatemi i seguenti volumi pagherò al postino l'importo indicato più le spese di spedizione.

☐ Pagamento anticipato senza spese di spedizione.

Nome ____________________

Cognome ____________________

Via ____________________ N. ______

Città ____________________ Cap. ______

Codice Fiscale ____________________

Data ____________ Firma ____________________

	Volume	Prezzo	Abbonati
N. ——	Audio Handbook	**L. 9.500**	(Abb. L. 8.550
N. ——	Manuale del Riparatore Radio-TV	**L. 18.500**	(Abb. L. 16.650
N. ——	SC/MP	**L. 9.500**	(Abb. L. 8.550
N. ——	Bugbook V	**L. 19.000**	(Abb. L. 17.100
N. ——	Bugbook VI	**L. 19.000**	(Abb. L. 17.100
N. ——	Timer 555	**L. 8.600**	(Abb. L. 7.750
N. ——	Bugbook I	**L. 18.000**	(Abb. L. 16.200
N. ——	Bugbook II	**L. 18.000**	(Abb. L. 16.200
N. ——	Bugbook II/A	**L. 4.500**	(Abb. L. 4.050
N. ——	Bugbook III	**L. 19.000**	(Abb. L. 17.100
N. ——	La Progettazione dei Filtri Attivi	**L. 15.000**	(Abb. L. 13.500
N. ——	La Progettazione degli Amp. Op.	**L. 15.000**	(Abb. L. 13.500

SOFTWARE

L'accesso su memorie di massa in BASIC

a cura della **Redazione**

Nel numero 2 di BIT sono state introdotte le istruzioni di I/0 e le principali tecniche di accesso alle informazioni memorizzate su cassette magnetiche o dischi (diskette). Questi concetti sono ora evidenziati tramite significativi e completi esempi.

Si supponga di voler costruire un programma che memorizza ricette su un nastro magnetico; queste ricette vengono ricercate per mezzo di una chiave numerica dedotta dagli alimenti usati nella preparazione di quel determinato piatto.
Le funzioni svolte da questo programma sono:
inserimento di una ricetta,
ricerca di una particolare ricetta,
stampa di tutte le ricette contenute nel file.
È evidente che queste tre funzioni verranno eseguite separatamente, e costituiscono parti di programma indipendenti per cui è accettabile che questi non risiedano contemporaneamente in memoria.
Gli alimenti sono divisi in quattro categorie, in base al loro valore nutritivo (Fig. 1).

(A) Farinacei	**(B) Proteiene**
1. Pane (farina)	1. Manzo o Maiale
2. Avena	2. Pollame
3. Riso	3. Pesce
4. Granoturco	4. Uova
5. Maccheroni	5. Fagioli
6. Taglierini	6. Noci
7. Spaghetti	7. Nessuno di questi
8. Nessuno di questi	

(C) Derivati del latte	**(D) Frutta e Verdura**
1. Latte	1. Agrumi
2. Formaggio	2. Melone
3. Nessuno di questi	3. Succhi di frutta
	4. Verdure
	5. Pomodori
	6. Nessuno di questi

Figura 1 - Corrispondenza tra ingredienti base, divisi per categorie, e indici semplici.

L'idea che sta dietro a questo sistema di ricerca è piuttosto semplice, ma il metodo usato per organizzarla è molto avanzato. Ad esempio la ricetta del Riso spagnolo che si prepara con riso (= 3), carne di manzo (= 1), niente derivati del latte (= 3), verdure (= 4), scritta in modo compatto nel seguente modo:
FOOD (3, 1, 3, 4) = Riso spagnolo
Pertanto gli alimenti *base* usati per preparare un piatto costituiscono le chiavi per ottenere in fase di ricerca il testo intero della ricetta memorizzata su nastro.

Programma ad overlay

Si supponga che il programma per memorizzare e ricercare le ricette sia troppo grande per poter essere contenuto nella memoria del calcolatore; per questo motivo è stato spezzato in segmenti chiamati *overlays*. Ciascun overlay viene accodato al programma principale quando è necessario.
L'accodamento si può ottenere in Basic in diversi modi. In questo caso, è usato il comando
APPEND = location, = step
dove:
location = numero dell'istruzione del programma principale da cui partire per accodare l'overlay;
step = passo di incremento usato per numerare le istruzioni successive.
Nel programma principale, riportato in Figura 2, le stringhe A$, B$, C$, D$, ed E$ vengono usate per me-

```
100 REM ***************************************
110 REM
120 REM         PROGRAMMA PRINCIPALE
130 REM
140 REM ***************************************
150 REM
160 DIM A$(72),B$(72),C$(72),D$(72),E$(72),
170 REM
180 FIND 6
190 APPEND 1000,10
200 REM
210 PRINT "LISTA OPERAZIONI:"
220 PRINT
230 PRINT "1. INSERIMENTO RICETTE 2.STAMPA TUTTE LE RICETTE"
240 PRINT "3. RICERCA UNA RICETTA 4. STOP"
250 PRINT
260 PRINT "SCEGLI OPERAZIONE (DA 1A 4)";
270 INPUT F
280 IF F = >4 THEN 520
290 REM
300 REM ------OVERLAY RELATIVA ROUTINE--------
310 REM
320 GOSUB F OF 370,420,470
330 GO TO 200
340 REM
350 REM --------------------------------------
360 REM
370 REM -----OVERLAY INSERIMENTO RICETTE-----
380 FIND 3
390 APPEND 2000,10
400 GOSUB 2000
410 GO TO 200
420 REM -----------STAMPA RICETTE-------------------
430 FIND 4
440 APPEND 2000,10
450 GOSUB 2000
460 GO TO 200
470 REM -----------RICERCA UNA RICETTA---------------
480 FIND 5
490 APPEND 2000,10
500 GOSUB 2000
510 GO TO 200
520 STOP
540 REM         AREA DI OVERLAY
1000 REM        ROUTINE DI INPUT
2000 REM        AREA DI OVERLAY
```

Figura 2 - Programma principale.

morizzare massimo cinque righe di testo della ricetta. Queste cinque righe vengono scritte su nastro, se si è nella fase di inserimento del testo, oppure, se si sta cercando una ricetta precedentemente scritta su nastro, in queste cinque righe è memorizzato il testo della ricetta, una volta trovato, che è quindi visualizzato.
Le linee 180 e 190 del programma principale (Fig. 2) mostrano come si apre un file su nastro (= 6) e come, dopo aver letto un sottoprogramma, lo si memorizza nelle locazioni 1000, 1010, 1020....Il sottoprogramma in questione è usato per introdurre gli alimenti base scelti per il menu.
Le linee 200 - 290 mostrano come l'utente può scegliere una delle quattro funzioni programmate.
La funzione 1 permette di introdurre il testo completo della ricetta che poi è memorizzata in un record su nastro.
La funzione 2 è usata per cercare una ricetta dati gli ingredienti base; la funzione 3 è usata per stampare tutte le ricette.
Dopo aver scelto la funzione, il programma salta ad uno dei tre comandi APPEND; il sottoprogramma attivato è caricato a partire dalla locazione 2000 del programma principale e l'esecuzione prosegue. Si noti come le locazioni 2000, 2010, 2020....(Fig. 2) vengano usate da tre *diversi* sottoprogrammi memorizzati sui files = 3, = 4, = 5. Questa è l'idea che sta dietro all'overlay: *segmenti diversi di programma usano le stesse locazioni di memoria in momenti diversi.*

Archivio su nastro

Si supponga ora di aver scelto la funzione 1; il comando APPEND alla linea 390 permette di caricare la routine di Figura 3. È opportuno analizzare attentamente questo sottoprogramma per vedere in che modo si legge un nastro e come si utilizza un processo chiamato *interrupt*.

```
2000 REM
2010 REM ************************************************
2020 REM
2030 REM      INSERIMENTO RICETTE SU FILE 2
2040 REM
2050 REM ************************************************
2060 REM
2070 REM ----------COSTRUZIONE FILE---------------
2080 PRINT "DIGITA CR QUANDO SEI PRONTO:";
2090 INPUT A$
2100 REM ----------RICERCA END OF FILE------------
2110 FIND 2
2120 ON EOF (0) THEN 2350
2130 INPUT @33:I,A$,B$,C$,E$
2140 GO TO TYP (0) OF 2170,2130,2130,2130
2145 GO TO 2170
2150 REM
2160 REM ----------SCRITTURA RICETTA--------------
2170 REM
2190 PRINT "VUOI AGGIUNGERE RICETTE? (SI/NO):";
2200 INPUT A$
2210 IF A$="NO" THEN 2340
2215 PRINT "IDENTIFICA RICETTA CON INGREDIENTI BASE"
2216 PRINT "DIGITA CR QUANDO SEI PRONTO";
2217 INPUT A$
2220 GOSUB 1000
2230 PRINT "DIGITA SOLO 5 LINEE, DI RICETTA"
2240 INPUT A$
2250 INPUT B$
2260 INPUT C$
2270 INPUT D$
2280 INPUT E$
2290 PRINT @33:I,A$,B$,C$,D$,E$
2300 REM
2310 PAGE
2320 GO TO 2170
2330 REM ----------ANCORA ??--------------------
2340 CLOSE @33:
2345 RETURN
2350 REM ************************************************
2360 REM
2370 REM      ROUTINE DI INTERRUZIONE PER EOF
2380 REM
2390 REM ************************************************
2400 REM
2410 OFF EOF (0)
2420 REM
2430 RETURN
2440 REM
```

Figura 3 - Overlay di inserimento testo.

Dopo aver aperto il file ≠2 (linea 2110) che contiene le ricette già memorizzate, con l'istruzione ON EOF (0) (linea 2120) è specificato che l'esecuzione normale del programma deve essere *interrotta* se la condizione di end-of-file (EOF) si presenta durante un'operazione di lettura del nastro.
In questo caso, il controllo passa alla *routine di interrupt (linea 2350); questa routine (linee 2350 - 2440) azzera la condizione di EOF e restituisce il controllo alla linea successiva a quella che a causato l'interrupt: in questo caso, alla linea 2140, essendo la linea 2130 (comando di lettura) quella che può far scattare la condizione di EOF.*
Le istruzioni 2130 e 2140 vengono eseguite ripetutamente finchè non è rilevato un EOF tramite interrupt, o tramite la funzione TYP(∅). (L'interruzione di EOF pone TYP(∅) ad 1). Se TYP(∅) ≠ 1, allora si salta alla linea 2170, diversamente alla linea 2130.
Questo modo di procedere permette di leggere un file già esistente sino in fondo, e di aggiungere in coda nuovi records, lasciando inalterati i preesistenti.
L'interrupt di EOF è necessario, in questo programma, in un'unica situazione: cioè quando il file è ancora vuoto.

Le istruzioni da 2170 a 2320 leggono ricette dal terminale e le scrivono (linea 2290) sul nastro (device ≠ 33). Quando non si vogliono più introdurre dati, viene chiuso il file sul device ≠ 33 e si rientra al programma principale.
Nel caso in cui si sia scelta la funzione 3, cioè la stampa di tutte le ricette, è utilizzato il sottoprogramma di Figura 4. Il file viene aperto (linea 2120), letto dall'inizio fino all'end-of-file, e il suo contenuto stampato

```
2000 REM
2010 REM ************************************************
2020 REM
2030 REM      STAMPA RICETTE DA FILE 2
2040 REM
2050 REM ************************************************
2060 REM
2070 PRINT "DIGITA CR QUANDO SEI PRONTO:";
2080 INPUT A$
2090 REM
2100 REM ----------LEGGI FILE 2 DALL'INIZIO----------
2110 REM
2120 FIND 2
2130 ON EOF (0) THEN 2360
2140 INPUT @33:I,A$,B$,C$,D$,E$
2150 GO TO TYP (0) OF 2350,2180,2180,2180
2160 GO TO 2350
2170 REM
2180 PRINT @Q: " RICETTA #",I
2190 PRINT @Q:
2200 PRINT @Q:
2210 PRINT @Q:
2220 PRINT @Q:A$
2230 PRINT @Q:B$
2240 PRINT @Q:C$
2250 PRINT @Q:D$
2260 PRINT @Q:E$
2270 PRINT @Q:
2280 PRINT @Q:
2290 PRINT @Q:
2300 PRINT @Q:
2310 PRINT @Q:
2320 GO TO 2140
2330 REM
2340 REM ----------RAGGIUNTO EOF---------------
2350 RETURN
2360 REM ************************************************
2370 REM
2380 REM ROUTINE DI INTERRUZIONE PER EOF
2390 REM
2400 REM ************************************************
2410 REM
2420 OFF EOF (0)
2430 RETURN
2440 REM
```

Figura 4 - Overlay di stampa testi.

sul device ≠ Q. Se ≠ Q identifica un terminale video, ogni ricetta sarà visualizzata sullo schermo; se invece ≠ Q identifica una stampante si ottiene la stampa della ricetta come in Figura 5.
Per la funzione 2 si usa invece il sottoprogramma di ricerca che è illustrato nella Figura 6; esso opera come tutte le routines di Input/Output su nastro.
La ricerca è fatta sequenzialmente dall'inizio alla fine del nastro; alla linea 2210 (Fig. 6) si confrontano I e J per vedere se le "chiavi" coincidono. In caso affermativo il record viene stampato sul device ≠ Q.
La routine di Input dei dati (routine che viene utilizzata nella linea 2220 di Figura 3 e nella linea 2090 di Figura 6) è illustrata nella Figura 7. Essa viene usata ogni volta che si vuole aggiungere una nuova ricetta su nastro e ogni volta che si ricerca una singola ricetta. Compito di questa routine è di fornire l'indice da associare ad un nuova ricetta o l'indice con cui numerare una ricetta già scritta, noti i quattro ingredienti principali scelti nell'elenco di Figura 1. Ad esempio l'indice del Bue alla Stroganoff è 648, ed è calcolato sapendo che dall'elenco (A) si sceglie l'ingrediente 6, dal (B) l'1 dal (C) il 3 e dal (D) il 6, cioè in breve (6, 1, 3, 6).

RICETTA = 648

BUE STROGANOFF - FAR ROSOLARE 1 CIPOLLA, 1 SPICCHIO D'AGLIO A FUOCO LENTO. UNIRE 1 KG. DI BUE, FAR COLORIRE. UNIRE 50 GR. DI FARINA, SALE, PEPE, 1/2 KG. DI FUNGHI. FAR CUOCERE 5 MIN. UNIRE 1/4 L. DI BRODO DI POLLO RISTRETTO. FAR BOLLIRE ADAGIO 10 MIN. UNIRE 1/4 L. DI PANNA ACIDA. ALZARE LA FIAMMA, COSPARGERE DI PREZZEMOLO. SERV. CON TAGLIERINI (4 - 6 PORZ.).

Figura 5 - Stampa di una ricetta.

Più esattamente, questa routine, dopo aver visualizzato le quattro categorie di ingredienti base e richiesto i relativi valori, che memorizza nelle variabili F, P, D e V (linee 1070 - 1435), passa al calcolo dell'indice I (linea 1460). I è espresso in funzione di F, P, D, V, cioè si ha una funzione di tipo HASH[1], in cui è assegnato un peso a ciascuna variabile.

```
2000 REM
2010 REM ************************************************
2020 REM
2030 REM      RICERCA RICETTE DA FILE 2
2040 REM
2050 REM ************************************************
2060 REM
2070 PRINT "DIGITA CR QUANDO SEI PRONTO:";
2080 INPUT A$
2090 GOSUB 1000
2100 REM
2110 REM ----------LEGGI FILE 2 DALL'INIZIO----------
2120 REM
2130 FIND 2
2140 ON EOF (0) THEN 2420
2150 INPUT @33:J,A$,B$,C$,D$,E$
2160 GO TO TYP (0) OF 2410,2190,2190,2190
2170 GO TO 2410
2180 REM
2190 REM ----------INDICI UGUALI?-----------------
2200 REM
2210 IF I=J THEN 2240
2220 GO TO 2150
2230 REM
2240 PRINT @0: "RICETTA #",J
2250 PRINT @0:
2260 PRINT @0:
2270 PRINT @0:
2280 PRINT @0: A$
2290 PRINT @0: B$
2300 PRINT @0: C$
2310 PRINT @0: D$
2320 PRINT @0: E$
2330 PRINT @0:
2340 PRINT @0:
2350 PRINT @0:
2360 PRINT @0:
2370 PRINT @0:
2380 GO TO 2150
2390 REM
2400 REM ----------RAGGIUNTO EOF----------
2410 RETURN
2420 REM ************************************************
2430 REM
2440 REM      ROUTINE DI INTERRUZIONE PER EOF
2450 REM
2460 REM ************************************************
2470 REM
2480 OFF EOF (0)
2490 PRINT "FINE FILE RICETTE"
2500 RETURN
2510 REM
```

Figura 6 - Overlay di ricerca testo.

La Figura 8 descrive come calcolare il "peso" per ogni variabile e come si arriva alla formula usata per il calcolo dell'indice: questo è necessario affinchè l'indice relativo a (6, 1, 3, 6) sia diverso ad esempio da quello ricavabile da (6, 1, 6, 3) o da (1, 6, 3, 6).
In questo modo al Bue alla Stroganoff è assegnato un solo numero, che contiene tutte le informazioni relative agli ingredienti necessari alla preparazione del piatto.
Il metodo descritto in questo esempio può essere utilizzato più vantaggiosamente su sistemi dotati di unità a disco anzichè a nastro; in questo caso l'indice I = 648 può essere usato per puntare in modo diretto al record che contiene la ricetta del Bue alla Stroganoff, senza bisogno quindi di cercare il record in modo sequenziale.

```
1000 REM ************************************************
1010 REM
1020 REM      INPUT INGREDIENTI BASE
1030 REM
1040 REM ************************************************
1050 REM
1060 PAGE
1070 PRINT "LISTA INGREDIENTI:"
1080 PRINT "  FARINACEI  "
1090 PRINT
1100 PRINT "1. PANE (FARINA)  2. AVENA"
1110 PRINT "3. RISO           4. GRANOTURCO"
1120 PRINT "5. MACCHERONI     6. TAGLIERINI"
1130 PRINT "7. SPAGHETTI      8. NESSUNO DI QUESTI"
1140 PRINT
1150 PRINT "SCEGLI (DA 1 A 8)";
1170 INPUT F
1180 PAGE
1190 PRINT "    PROTEINE"
1200 PRINT
1210 PRINT "1. MANZO               2. POLLAME"
1220 PRINT "3. PESCE               4. UOVA"
1230 PRINT "5. FAGIOLI             6. NOCI"
1240 PRINT "7. NESSUNO DI QUESTI
1250 PRINT
1260 PRINT "SCEGLI (DA 1 A 7)";
1270 INPUT P
1280 PAGE
1290 PRINT "  DERIVATI DEL LATTE"
1300 PRINT
1310 PRINT "1. LATTE  2. FORMAGGIO  3. NESSUNO DI QUESTI"
1320 PRINT
1330 PRINT "SCEGLI (DA 1 A 3)";
1340 INPUT D
1350 PRINT
1360 PRINT "      FRUTTA E VERDURA"
1370 PRINT
1380 PRINT "1. AGRUMI                  2. MELONE"
1390 PRINT "3. SUCCHI DI FRUTTA   4. VERDURE"
1400 PRINT "5. POMODORI                6. NESSUNO DI QUESTI"
1410 PRINT
1420 PRINT "SCEGLI (DA  1 A 6)";
1430 INPUT V
1435 REM
1440 REM -----CALCOLA NUMERO DI IDENTIF---------
1450 REM
1460 LET I= 126*F+18*P+6*D+V-150
1480 RETURN
1500 REM
```

Figura 7 - Overlay di Input dati.

Archivio su unità a disco

Le operazioni di lettura e scrittura di un file su disco possono essere fatte sia in modo *sequenziale* sia in modo *random* o *diretto*.
In Figura 9 sono illustrate le routines di lettura e scrittura *sequenziale*.
I dati vengono presi dalla stringa A$ e messi nel file sequenziale chiamato DATA; l'istruzione OPEN ini-

[1] *cfr. BIT n.2*

```
FMAX = 8, PMAX = 7, DMAX = 3, VMAX = 6

I = PESOf*(F-1)
  + PESOp*(P-1)
  + PESOd*(D-1)
  + PESOv*(V-1)

PESOf = PMAX * DMAX * VMAX          = 126
PESOp = DMAX * VMAX                 =  18
PESOd = VMAX                        =   6
PESOv = 1                           =   1
I = 126*(F-1) + 18*(P-1) + 6* (D-1) + V =
  = 126*F + 18*P + 6*D + V - 150.
```

Figura 8 - Algoritmo di calcolo di una funzione HASH.

zializza il file logico =1 ad accettare la scrittura (lettura) sul file fisico DATA. La funzione TYP(1) viene stampata per mostrare quali valori assume in quanto tali valori possono semplificare la programmazione dell'Input/Output: infatti, se il record successivo a quello letto è un end-of-file, TYP(1) vale ∅, altrimenti TYP(1) = 1 (Nella routine di scrittura, EOF vale sempre zero, poichè ciascuna WRITE memorizza un record su disco e fa si che venga scritto un EOF dopo il record).

È possibile leggere e scrivere un record *in modo diretto* inserendo % K nel comando READ/WRITE:

READ = 1%K, A, B
WRITE = 1%K, A, B

Questi due comandi fanno si che si acceda al record K-mo (iniziando a contare da zero) del file=1 e si leggano o scrivano i valori di A e B.

Per evidenziare come, nel caso di disco, è possibile accedere ad un dato record senza aver bisogno di fare una ricerca sull'intero archivio di records, è opportuno che i dati siano organizzati sul supporto in maniera particolare. Supponendo ad esempio che ogni record dell'archivio contenga i dati anagrafici di una persona più altre informazioni (ad es. numero telefonico, componenti nucleo familiare, occupazione,

```
(A) ROUTINE DI SCRITTURA SEQUENZIALE
100   REM ************************************************
110   REM
120   REM      ROUTINE DI SCRITTURA FILE
130   REM
140   REM ************************************************
150   REM
160   DIM A$(80)
170   OPEN # 1,"DATA"
180   FOR I = 1 TO 10
190      INPUT "ENTRA IL TUO NOME:",A$
192      IF A$ = "  " THEN END
194      WRITE # 1,A$
200      PRINT TYP (1)
210   NEXT I
220   REM ---------------------------
230   CLOSE # 1
240   PRINT "FINE"
250   END

(B) ROUTINE DI LETTURA SEQUENZIALE
100   REM ************************************************
110   REM
120   REM      ROUTINE DI LETTURA FILE
130   REM
140   REM ************************************************
150   REM
160   DIM A$(80)
170   OPEN # 1,"DATA"
180   FOR I = 1 TO 10
190      READ # 1,A$
200      PRINT A$
205      PRINT TYP(1)
210   NEXT I
220   REM ---------------------------
230   CLOSE # 1
240   PRINT "FINE"
250   END
```

Figura 9 - Routines di scrittura/lettura sequenziale per Disco.

etc.), si organizzino i records secondo una struttura *ad albero* in cui il cognome è la chiave con cui si percorre l'albero delle chiavi che, a sua volta, è costruito secondo l'ordine alfabetico. Pertanto, quando si guarda la *radice* dell'albero (Figura 10), si nota che tutto ciò che si trova a sinistra di "GAMBINI" viene prima (alfabeticamente) di "GAMBINI", mentre a destra di "GAMBINI" si trovano cognomi che vengono dopo in ordine alfabetico.

Questo ordinamento si mantiene ad ogni livello dell'albero, e ciò permette di individuare un singolo record scartando ad ogni passo tutto ciò che si trova in una delle due direzioni di ramificazione.

Volendo, ad esempio, cercare il record relativo a "DURANTE", si inizi dalla radice dell'albero che è il record ∅ dell'archivio, convenzionalmente chiamato INTESTAZIONE. Questo contiene due *puntatori* D ed S (destro e sinistro). Il puntatore S non è utilizzato, per cui seguendo il puntatore D si accede al record 2 relativo a "GAMBINI", tramite l'istruzione:

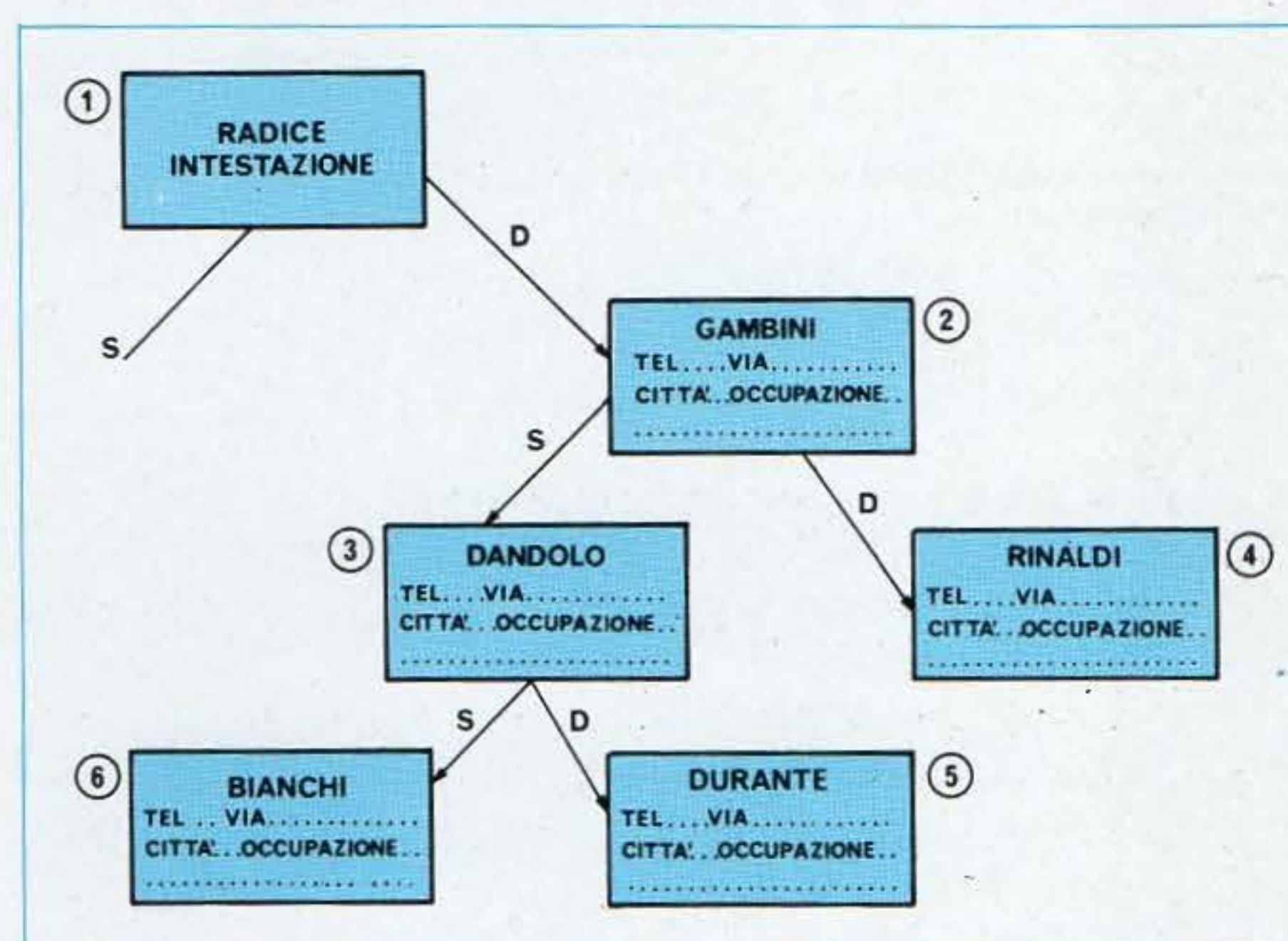



Figura 10 - Albero di Data Base. Il numero del record è nel circolo. La chiave è in testa ad ogni riquadro.

READ =1 %L*N + H, J$, A$, B$, C$, S, D

in cui: L = lunghezza di ogni record
H = lunghezza del record INTESTAZIONE
N = numero del record

Essendo la chiave del record 2 diversa da "DURAN-

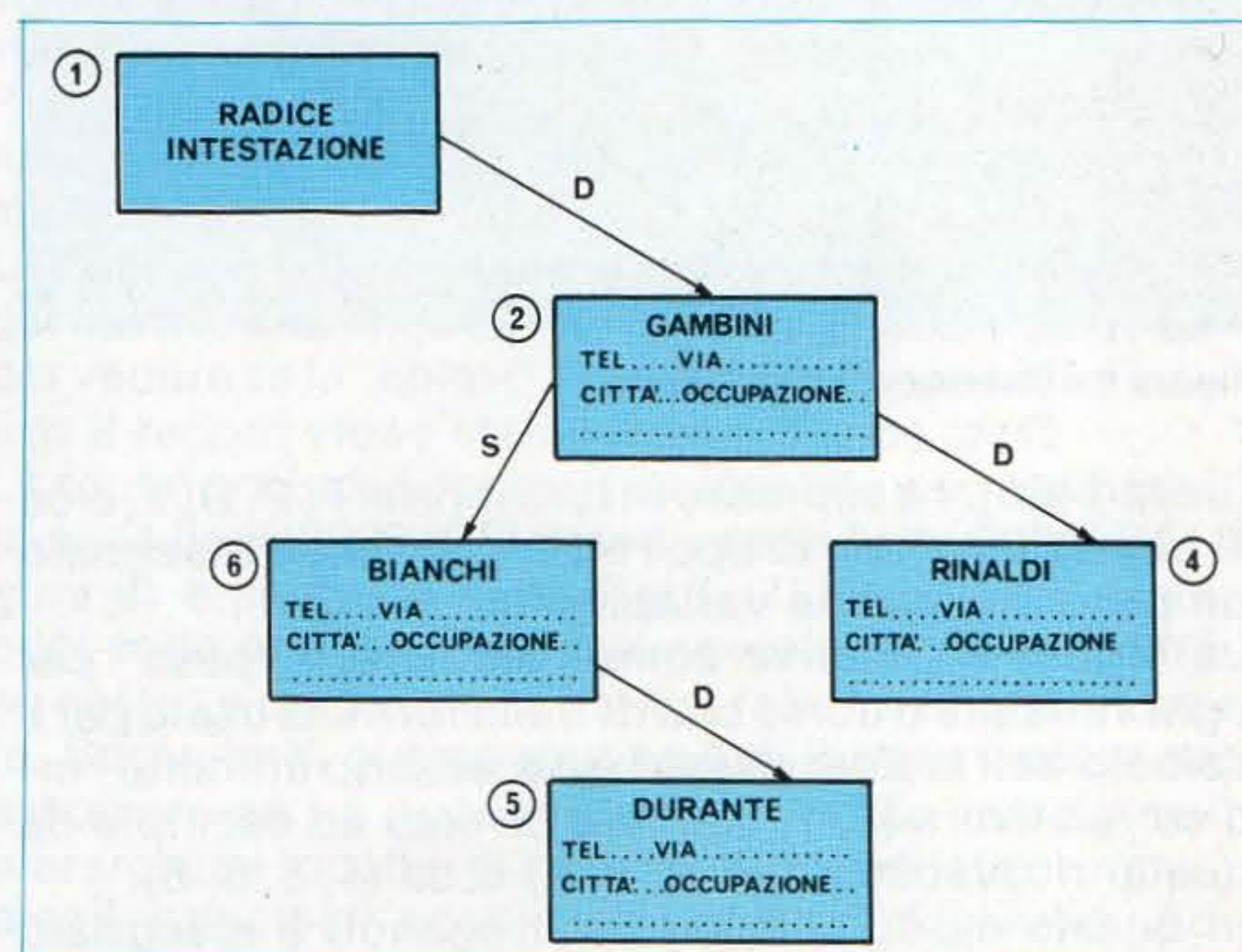



Figura 11 - Configurazione dell'albero di Data Base dopo l'annullamento di un record.

1 RADICE INTESTAZIONE
D
2 GAMBINI TEL....VIA.......... CITTA'..OCCUPAZIONE..
S
D
6 BIANCHI TEL....VIA........... CITTA'..OCCUPAZIONE..
4 RINALDI TEL....VIA........... CITTA'..OCCUPAZIONE..
D
5 DURANTE TEL....VIA........... CITTA'..OCCUPAZIONE..
S
3 DANDOLO TEL....VIA........... CITTA'..OCCUPAZIONE..

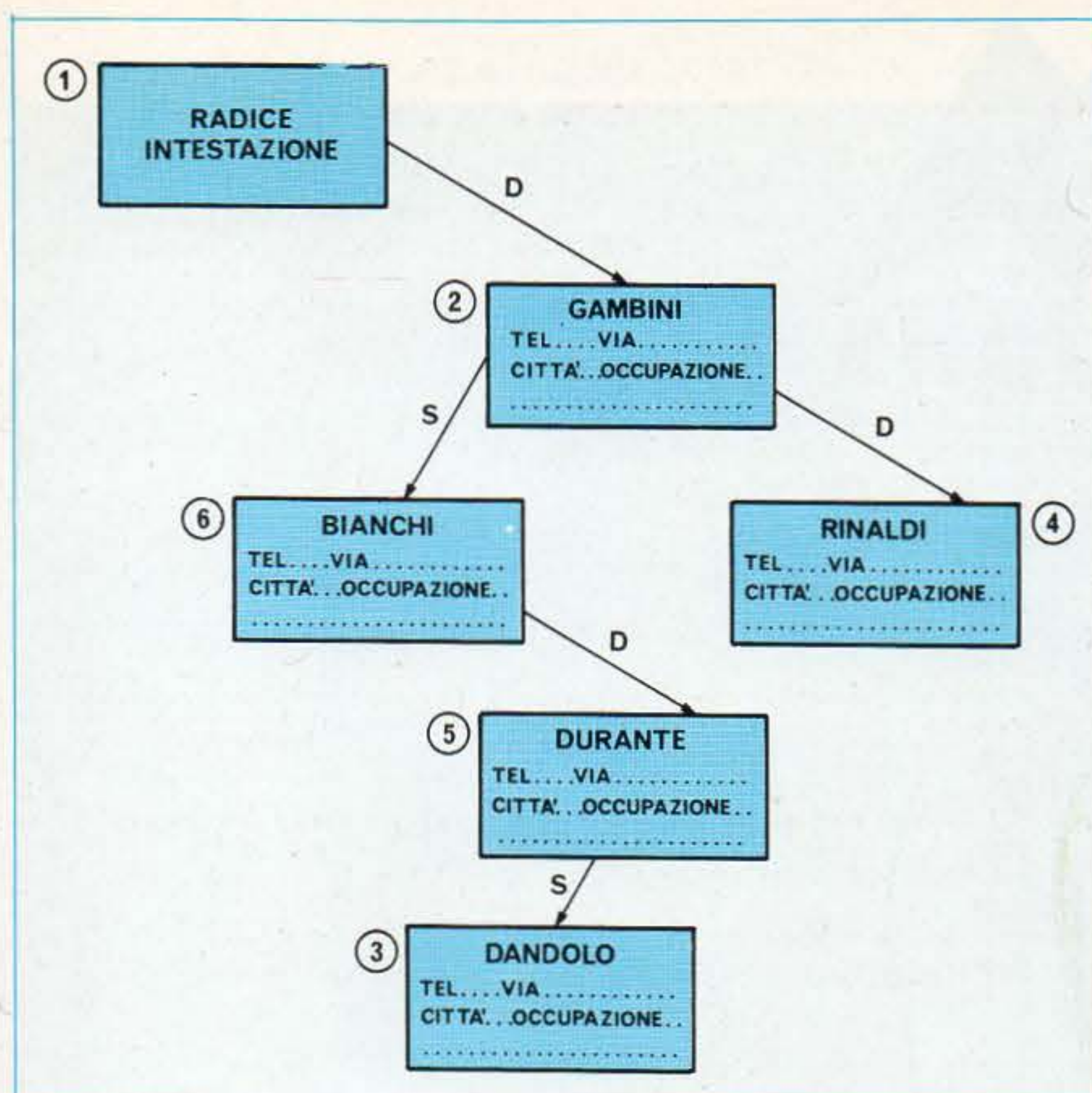



Figura 12 - Configurazione dell'albero di Data Base dopo l'inserimento di un record.

TE", si consideri il puntatore S, in quanto "DURANTE" è *minore* di "GAMBINI".
Si ottiene in tal modo il record relativo a "DANDOLO", che genera un nuovo cammino. In questo caso verrà scelto il puntatore D, essendo "DURANTE" > "DANDOLO". Il record così ottenuto è questa volta quello voluto: la ricerca è stata ottenuta con tre confronti invece che con cinque (caso della ricerca sequenziale).
L'albero del *Data Base* deve essere mantenuto ordinato anche quando vengono rimossi o inseriti dei records; in Figura 11 si può vedere ciò che si ottiene dopo aver cancellato il record "DANDOLO" dall'albero di Figura 10.
Quando si cancella un record, lo si sostituisce con il suo subrecord di sinistra; il subrecord di destra è invece attaccato al ramo più a destra del subrecord di sinistra.
Nell'esempio di Figura 11 il record con chiave "DURANTE" è pertanto attaccato semplicemente come subrecord di destra di "BIANCHI".

Quando un nuovo record è inserito, è necessario fare attenzione a mantenere ordinata la struttura che se ne ricava. La Figura 12 mostra la nuova configurazione dell'albero quando si inserisce il record "DANDOLO": questo diventa il subrecord di sinistra di "DURANTE".
In Figura 13 si ha un insieme di programmi che fanno riferimento all'albero di Data Base descritto.
Si noti fra l'altro il metodo usato per allocare lo spazio sul disco mediante un *vettore di bits* F$ (stringa di zeri e uni), ciascuno dei quali indica se il record nella posizione corrispondente è presente o no sul disco. Se F$ (A1) è zero, è possibile usare il record nella posizione A1 per memorizzare una nuova voce dell'albero.
F$ (I, J) significa estrarre da F$ la sottostringa che comincia con il bit nella posizione I e termina con il bit nella posizione J.
Le routines di assegnazione e cancellazione operano su F$ per occupare records quando è necessario e per "restituirli" a F$ quando sono cancellati dal file.
La routine di inizializzazione costruisce un record di informazioni relative al file; questo record salva il valore di F$, il nome del file, la lunghezza L (in caratteri) di ogni record, e la lunghezza H (in caratteri) del record di INTESTAZIONE.
L'inizializzazione viene fatta una sola volta per ogni applicazione.
La routine di ricerca opera sul file usando la ricerca sull'albero di Data Base descritta prima.
La ricerca ha termine quando la chiave da ricercare (K$) è uguale alla chiave del record memorizzato (J$).
I programmi di Figura 13 ricercano un posto disponibile seguendo l'albero; il nuovo record verrà aggiunto all'albero solo se non è già presente una chiave uguale.
L'albero di Data Base è generale e può essere usato in svariate applicazioni, soddisfacendo sempre le condizioni:

1) Ricerca rapida
2) Possibilità di aggiungere nuove informazioni in un momento qualsiasi
3) Possibilità di cancellare vecchi records in un momento qualsiasi
4) Possibilità di cambiare rapidamente una voce in un momento qualsiasi. ■

```
100 REM **************************************************
110 REM
120 REM     PROGRAMMI DI DATA BASE
130 REM
140 REM **************************************************
150 REM
155 LET M= 20 :  REM -----MAX. LUNGH. DI F
160 DIM F$(20),P$(8),O$(8),A$(20),B$(20),(20),D$(20),J$(12),K$(12)
165 INPUT "ENTRA LA PAROLA - CHIAVE (8 CAR.):",P$
170 PRINT "SELEZIONA UNA DELLE FUNZIONI:"
 75 PRINT
180 PRINT "     1. INIZIALIZZA IL DATABASE"
185 PRINT "     2. AGGIUNGI UNA VOCE AL DATABASE"
190 PRINT "     3. LEGGI UNA VOCE DAL DATABASE"
195 PRINT "     4. CANCELLA UNA VOCE DAL DATABASE"
200 PRINT "     5. STOP."
210 INPUT " DIGITA LA FUNZIONE SCELTA:",N
220 IF N> 5 THEN 170
225 IF N< 1 THEN 170
230 REM
235 REM -----------ESECUZIONE ROUTINE SCELTA----------
240 REM
250 ON N GOTO 251,252,253,254,298
251 GOSUB 1000  :  GOTO  270
252 GOSUB 2000  :  GOTO  270
253 GOSUB 3000  :  GOTO  270
254 GOSUB 4000  :  GOTO  270
260 REM
270 CLOSE # 1
275 OPEN  # 1, P$
280 WRITE # 1 %0,P$,F$,S,H
290 CLOSE # 1
295 GOTO  170
296 REM
298 CLOSE # 1
299 END
300 REM **************************************************
310 REM
320 REM            ALLOCA SPAZIO
330 REM
340 REM **************************************************
350 IF  F$ (M,M)= "1" THEN 355 ELSE 365
355 PRINT " DISCO PIENO"
360 LET  A1=0 : RETURN
361 REM
365 FOR  I=1  TO  M
370 IF   F$ (I,I)= "1" THEN 385 ELSE 375
375 LET  A1=I
380 EXIT  390
385 NEXT  I
390 LET  F$ (A1,A1)= "1" : RETURN
400 REM **************************************************
410 REM
420 REM            LIBERA SPAZIO
```

```
430 REM
440 REM ************************************************
450 REM
460 LET F$ (A1,A1)="0"
465 PRINT "RECORD", A1, "CANCELLATO"
470 RETURN
600 REM ************************************************
601 REM
602 REM          RICERCA SULL'ALBERO DEL DATABASE
603 REM
604 REM ************************************************
605 REM
606 REM      A2  E' IL NODO TERMINALE DELL'ALBERO DOPO LA RICERCA
607 REM      A4 E' Ø SE "NON TROVATO", A4=A2 SE "TROVATO"
608 REM      A3 PUNTA AL NODO PRECEDENTE A2
609 REM
610 REM -------RIPETI FINCHE' IL NODO TERMINALE NON E' TROVATO---
611 REM
620 LET   A4 =1 :  LET  A2=0
630 FOR   I =1  TO  M
640  READ  # 1 % A4*S+NJ$,C$,D$,L,R
650 LET  A3 =A2  :  LET A2=A4
655 REM
656 REM ------VERIFICA EGUAGL.; SE EG., "TROVATO"-----
657 REM
660   IF  J$=K$  THEN 735
670   IF  J$<K$  THEN 680 ELSE 700
680 LET  C=1  :  LET A4=R
690 GOTO 720
695 REM
696 REM -----COND. ELSE: SUBALBERO DI SINISTRA-----
698 REM
700 LET  C=0  :  LET  A4=L
710 REM -----FINE DELLA ISTR. IF - THEN - ELSE-----
720 IF  A4=0  THEN 735
730 NEXT I
735 REM
740 RETURN
1000 REM ************************************************
1001 REM
1002 REM        INIZIALIZZA DATA BASE
1003 REM
1004 REM ************************************************
1005 REM
1010 OPEN # 1,P$
1015 REM
1016 REM ----------DAI LUNG. RECORD-----------------------
1017 REM
1020 INPUT " ENTRA LUNG. RECORD IN CARATTERI :",S
1025 IF  S>255  THEN 1020
1030 INPUT " ENTRA LUNG. INTEST. IN CARATTERI :",N
1035 IF N >255 THEN 1030
1040 REM
1041 REM ----------INIZIALIZZA TABELLA ALLOCAZIONE----------
1042 REM
1045 LET  F$(1,1)="1"
1050 FOR  I=2  TO  M
1055  LET  F$(I)="0"
1060 NEXT  I
1070 REM
1075 REM ----------SCRITTURA RADICE DELL'ALBERO-------------
1079 REM
1080 LET   J$="%$#?!&!*$%&$" :C$="RECORD INTEST.":D$"ALBERO DI RIC
1090 WRITE  #1  %S+N,J$,C$,D$,0,0
1100 RETURN
1120 REM
2000 REM ************************************************
2001 REM
2002 REM             SCRITTURA SU DATA BASE
2003 REM
2004 REM ************************************************
2005 REM
2010 OPEN #1,P$
2020 READ #1  %0,Q$,F$,S,N
2030 IF  P$=Q$ THEN 2060
2040 PRINT  " PAROLA CHIAVE ERRATA!"
2050 RETURN
2055 REM
2060 INPUT "ENTRA NOME (12 CAR. MAX.) :",K$
2070 IF K $= "" THEN RETURN
2075 REM
2080 INPUT "ENTRA ATTRIBUTI (20 CARAT. MAX.) :",B$
2090 INPUT "ENTRA ULT.ATTRIBUTI (20 CARAT. MAX.) :",D$
2100 REM
2105 REM ----------E' QUESTO IL PRIMO RECORD?-----------
2106 REM
2110 IF  F$ (1,1)="0" THEN 2120 ELSE 2200
2120 LET  F$ (1,1)="1"
2130 REM
2134 REM ----------SI, INSERISCI NUOVO RECORD-----------
2136 REM
2140 WRITE  #1 %S+N,K$,A$,B$,L,R
2150 GOTO  2060
2160 REM
2170 REM ---------NO, VERIFICA SPAZIO NELL'ALBERO--------
2180 REM
2200 GOSUB 600
2210 REM
2220 IF  A4=0 THEN 2274
2260 PRINT "RECORD ESISTENTE!!!"
2270 GOTO 2060
2271 REM
2272 REM ----------OCCUPA SPAZIO------------------------
2273 REM
2274 GOSUB  300
2275 REM
2276 REM ---------AGGIORNA PUNTATORE---------------------
2277 REM
2280 READ  #1 %A2*S+N,J$,C$,D$,L,R
2285 REM
2290 IF  C=0 THEN L =A1  ELSE  R=A1
2300 WRITE  #1 %A2*S+N,J$,C$,D$,L,R
2310 WRITE #1 %A1*S+N,K$,A$,B$,0,0
2320 GOTO 2060
2400 REM
2405 REM -----SCRITTURA NUOVO RECORD SUL DATA BASE ESEGUITA
2406 REM
2500 RETURN
2510 REM
3000 REM ************************************************
3001 REM
3002 REM              LETTURA RECORD
3003 REM
3004 REM/************************************************
3005 REM
3010 OPEN #1,P$
3015 READ #1 %0,Q$,F$,S,N
3016 IF  P$=Q$  THEN 3020
3017 PRINT "PAROLA - CHIAVE ERRATA !!!"
3018 RETURN
3019 REM
3020 INPUT " ENTRA IL NOME :,K$
3030 IF  K$="" THEN RETURN
3040 REM
3041 REM ----------VERIFICA NOME NELL'ALBERO-------------
3042 REM
3050 GOSUB 600
3055 IF  A4=0 THEN 3100
3056 REM
3057 REM ----------RECORD TROVATO!-----------------------
3058 REM
3060 PRINT "NOME :",J$
3061 PRINT "DATI ANAGRAFICI: ",C$
3062 PRINT "OCCUPAZIONE:",D$
3070 REM
3071 REM ---------DESIDERI ALTRI RECORDS?-----------------
3072 REM
3080 GOTO 3020
3085 REM
3090 REM ----------NOME ERRATO----------------------------
3095 REM
3100 PRINT "NON ESISTE IL NOME CHIESTO !"
3110 GOTO 3020
3115 REM
3201 REM
4000 REM ************************************************
4001 REM
4002 REM            CANCELLAZIONE RECORD
4003 REM
4004 REM ************************************************
4050 REM
4055 OPEN #1,P$
4056 READ #1 %0,Q4,F$,S,N
4060 INPUT "ENTRA IL NOME DA CANCELLARE:",K$
4070 IF  K$=""  THEN RETURN
4080 REM
4085 REM ---------RICERCA RECORD--------------------------
4090 REM
4100 GOSUB 600
4110 REM
4120 IF  A4=0  THEN PRINT "NOME INESISTENTE."
4121 IF  A4=0  THEN 4060
4125 REM
4126 REM ----------NOME TROVATO--------------------------
4127 REM
4130 INPUT "CONFERMI OPERAZ.?",Q$
4140 IF  Q$ = "SI" THEN 4200
4150 PRINT "RIPETI RICHIESTA!"
4160 GOTO 4060
4170 REM
4171 REM ----------CANCELLAZIONE RECORD------------------
4172 REM
4200 READ #1 %A2*S+N,J$,C$,D$,L1,R1
4215 READ #1 %A3*S+N,J$,C$,D$,L,R
4216 IF  C=0  THEN L=L1 ELSE  R=L1
4220 WRITE  #1  %A3*S+N,J$,C$,D$,L,R
4240 REM
4241 REM -----LIBERA SPAZIO------------------------------
4242 REM
4250 LET  A1=A2
4270 GOSUB 400
4275 REM
4280 REM ----------VERIFICA SPAZIO PER INSERIRE R1-----------
4285 REM
4290  GOSUB 600
4295 REM
4300 READ #1 %A2*S+N,J$,C$,D$,L,R
4310 WRITE #1 %A2*S+N,J$,C$,D$,L,R1
4320 REM
4330 GOTO 4060
4340 REM
9999 REM
```

Cobol

Fortran

500

100

500

100

Pascal

Les jeux sont faits

Note introduttive su PASCAL

di **Ivan Maffezzini** - *Honeywell Information System Italia*

Alcune considerazioni su un linguaggio strutturato che razionalizza la progettazione del software.

"Parlare sempre non pensarci mai"(!?)
T.W. Adorno

Premessa

Non sarà certamente uno solo il lettore che, dopo un'occhiata al sottotitolo ed un sospirato "ancora!", passerà ad un altro articolo. Volutamente e con leggero desiderio di provocazione si è inserito quello "strutturato". Da un lato i legami tra il linguaggio PASCAL e la strategia di programmazione che va sotto il nome di programmazione strutturata (P.S.) sono talmente forti che non è possibile sottacerli; dall'altro la parola struttura con tutti i suoi derivati, ha, nel campo dell'informatica, la capacità, propria di ogni termine inflazionato, di dividere in due schiere contrapposte.
Considerando in particolare la situazione italiana si vede come lo schierarsi con uno dei due fronti non è, sostanzialmente, dovuto a scelte o idiosincrasie personali ma al fatto di appartenere al mondo dell'industria o a quello universitario. Trascurando il problema politico dell'interazione tra questi due mondi, dovremmo forse schermarci dietro diffusi luoghi comuni affermando che fanno solo giochi teorici senza utilità gli uni e sono solo dei maneggioni gli altri?

Certamente no. Ma questo, oltre che un 'no', vuole essere un invito agli addetti ai lavori a spogliare, per quanto è possibile, la parola 'struttura' dalle ormai prevalenti connotazioni emotive ed a valutare se è davvero *cattiva* la posizione di chi sostiene che la P.S. non riesce a sfondare nell'industria in quanto, in un campo dove, fino a non molto tempo fa, sembrava non servisse altro know-how che un buon senso da praticoni, il management sia composto soprattutto da persone le cui competenze tecniche non vanno al di là di un più o meno buon naso.[1]
Questa ultima presa di posizione polemica ci è servita e per chiarire la nostra posizione e per togliere di mezzo l'ingombrante 'struttura'.

Origini e sviluppo

Una versione preliminare di PASCAL nasce, per opera di un gruppo che fa capo a N. Wirth, presso l'istituto di informatica della università di Zurigo nel '68. Dopo alcune revisioni, nel '73, viene pubblicato il rapporto che ne fissa gli standards. I motivi che hanno spinto Wirth ad implementare un nuovo linguaggio sono spiegati all'inizio del rapporto:
". . . rendere disponibile un linguaggio adatto all'insegnamento della programmazione come disciplina sistematica . . . sviluppare implementazioni che siano affidabili ed efficienti . . ."

Considerata la diffusione dei compilatori PASCAL, si vede come la necessità di affidabilità e sistematicità non fosse sentita solo nel mondo dell' insegnamento: esistono infatti, attualmente, compilatori per una settantina di elaboratori che vanno da un micro come 8080 ad un maxi come il 6000 Honeywell. Molto intenso è stato il lavoro nel campo dei piccoli sistemi; in particolare presso l'università di S. Diego in California è stato sviluppato un sistema -UCSD PASCAL- con alcune estensioni rispetto allo standard ed una alta portabilità il quale è disponibile con listings e documentazione per una cifra dell'ordine di $ 200.[2] Per garantire una alta portabilità, tutto il s/w UCSD, eccettuato l'interprete, è scritto in PASCAL: il compilatore genera infatti un codice intermedio (P-code) che fa riferimento ad una macchina virtuale la quale viene, via interprete, emulata sulla macchina reale. La necessità dell'interprete rallenta la macchina di 5-6 volte; è comunque degno di nota che la Western Digital ha superato il problema della velocità realizzando in h/w (5 integrati) un processore che esegue direttamente il P-code.

PASCAL ha trovato applicazione nei campi più svariati: non solo si è dimostrato ottimo nella progettazione di sistemi e nel calcolo scientifico ma, anche nelle applicazioni commerciali, lasciando la parola a Bowles: *". . . le sue facilitazioni sono di molto superiori a quelle del* COBOL".
Al di là del tipo di linguaggio usato nell'implementazione, PASCAL, con regole sintattiche più o meno rigidamente applicate, è ormai diffusamente impiegato nella progettazione del s/w.

Progettazione

La P.S. e le metodologie di progettazione di sua filiazione sono nate quando l'analista si è trovato di fronte a problemi *complessi*.

---

[1] *Per contro i dirigenti di buon naso potrebbero, rovesciando la medaglia, vedervi inscritta la voglia di giovani tecnocrati di scannarsi per la illusione di un po' di potere.*

[2] *Per ottenere sistema e documentazioni su dischetto o disco scrivere a :*
Kenneth Bowles IIS University of California S. Diego La Jolla CA92093 USA

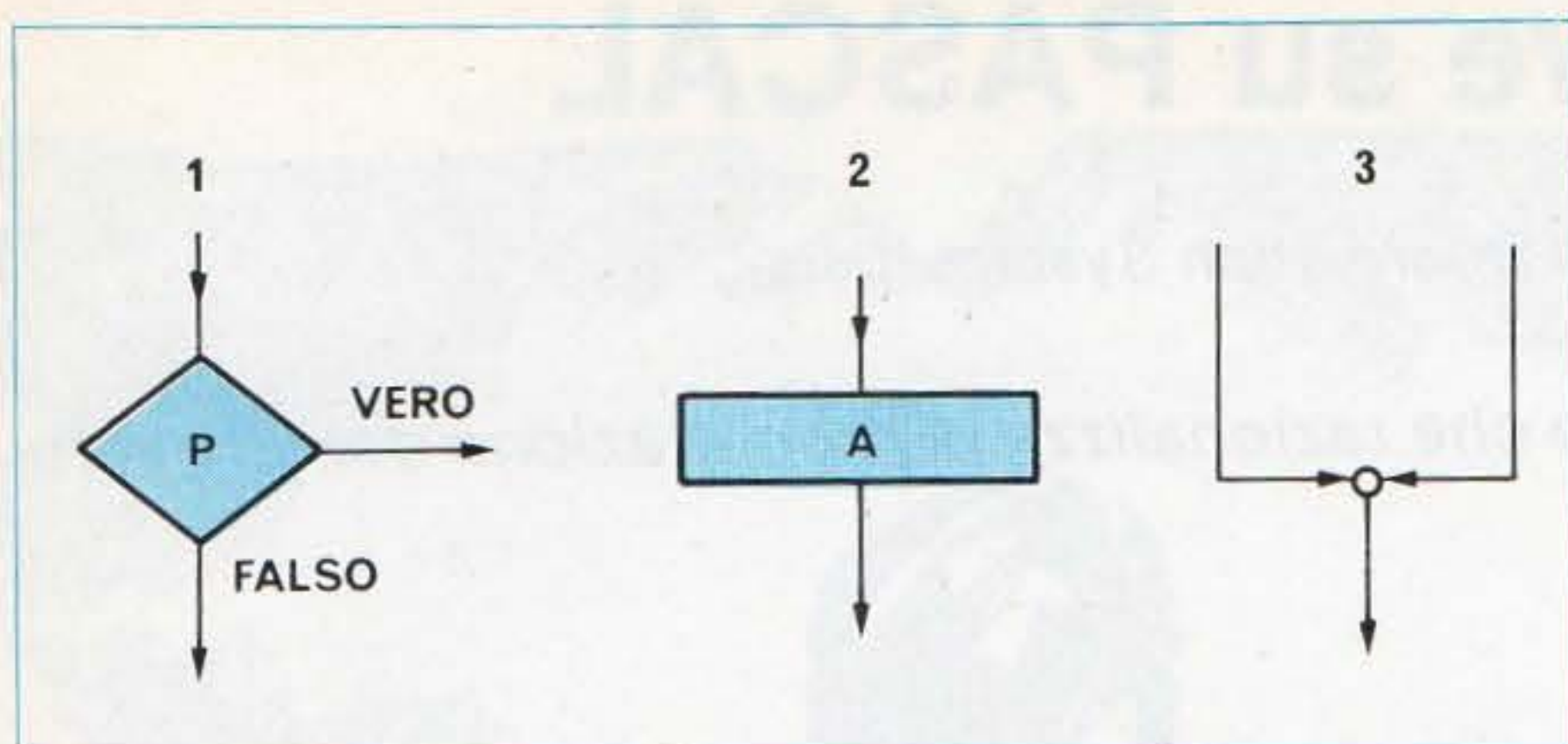



Figura 1 - 1) Blocco condizionale, 2) Blocco decisionale, 3) Punto di raccolta.

Per valutare la complessità di un programma va considerata non solo la difficoltà logica, *profondità*, ma anche la quantità di dati ed il numero di operazioni da eseguire su di essi, *estensione*.

Prima dell'avvento della P.S. era la dimensione della profondità che riceveva la maggiore attenzione: di qui i miti delle scorciatoie, dei risparmi, dell'originalità; si era sottovalutata la difficoltà della mente umana nel considerare grosse quantità di informazioni e questo soprattutto a causa della trappola tesa da una fuorviante analogia tra cervello umano[3] ed artificiale.

Quando (nell'industria!) ci si accorse che un prodotto s/w, più che lusingare la vanità del programmatore per il trucco che permetteva un risparmio di 19 bits, doveva essere affidabile, manutenibile, modificabile, in sostanza *vendibile*, si cominciò a parlare di ingegneria del s/w, di standards; si cominciò ad operare una netta distinzione tra costruzione (implementazione) del s/w e sua progettazione, ovvero ad *introdurre metodologie*. A questo punto è con sommo piacere che invitiamo il lettore dell'altro campo, che non ha ancora desistito, a chiudere la rivista perché è in arrivo un altro termine superinflazionato: TOP-DOWN.

In effetti una delle tecniche che meglio s'accoppiano con la P.S. è la tecnica di lavoro Top-down con la quale la struttura di PASCAL s'adatta perfettamente. Certamente né P.S. né Top-down né PASCAL sono la panacea nel s/w; sono comunque, a diversi livelli di astrazione, tre strumenti che aiutano a risolvere con chiarezza un problema.

Non è certo casuale se questi strumenti hanno reso il flow-charting un esercizio da nostalgici. Se il flow-charting può facilmente trasformarsi in un accatastamento di blocchi (fig. 1), la progettazione usando PASCAL facilita la costruzione di strutture

Figura 2 - Flow dei costrutti base di Pascal. 1) while P do A 2) if P then A else B 3) begin...A; B...end.

complesse usando i tipi di fig. 2. In effetti PASCAL ha avuto un'enorme diffusione come linguaggio di progettazione anche per il fatto che è poi molto facile passare ad un'implementazione in ASSEMBLER od in un altro linguaggio. Non è poi senza importanza il fatto che l'immediatezza visiva dei flow (loro caratteristica fondamentale) può essere addirittura migliorata in PASCAL sfruttando i capoversi (*Indentation*).

Diamo ora un rapido sguardo al linguaggio invitando il lettore non completamente digiuno a consultare il manuale di Wirth citato in bibliografia, il quale unisce una chiarezza esemplare ad una sintesi invidiabile; al neofita si consiglia il testo di Bowles.

Struttura dei programmi

Un programma in PASCAL è composto, oltre che da un'intestazione che specifica il nome ed i flussi su cui lavora, dalle sei parti, non tutte necessarie, sottoelencate:

1. Dichiarazione delle etichette
2. Definizione delle costanti
3. Definizione dei tipi
4. Dichiarazione delle variabili
5. Dichiarazione delle procedure e funzioni
6. Statements

Per una definizione formale della struttura di un programma si veda il diagramma sintattico di Fig.3. Dopo una breve analisi delle parti sopracitate cercheremo di mostrare, per mezzo di un esempio, l'uso di PASCAL sia per progettare il programma che per implementarlo.

1.-dichiarazione etichette

Nel caso in cui non si usano etichette nella parte 6, questa dichiarazione può mancare: ogni etichetta va dichiarata prima di essere usata.

Necessariamente nel caso di salti incondizionati (**goto**) ci vorranno delle etichette, ma come dice Jensen: "*La presenza di* **goto** *in un programma PASCAL è spesso indice che il programmatore non ha ancora imparato a pensare in* PASCAL".

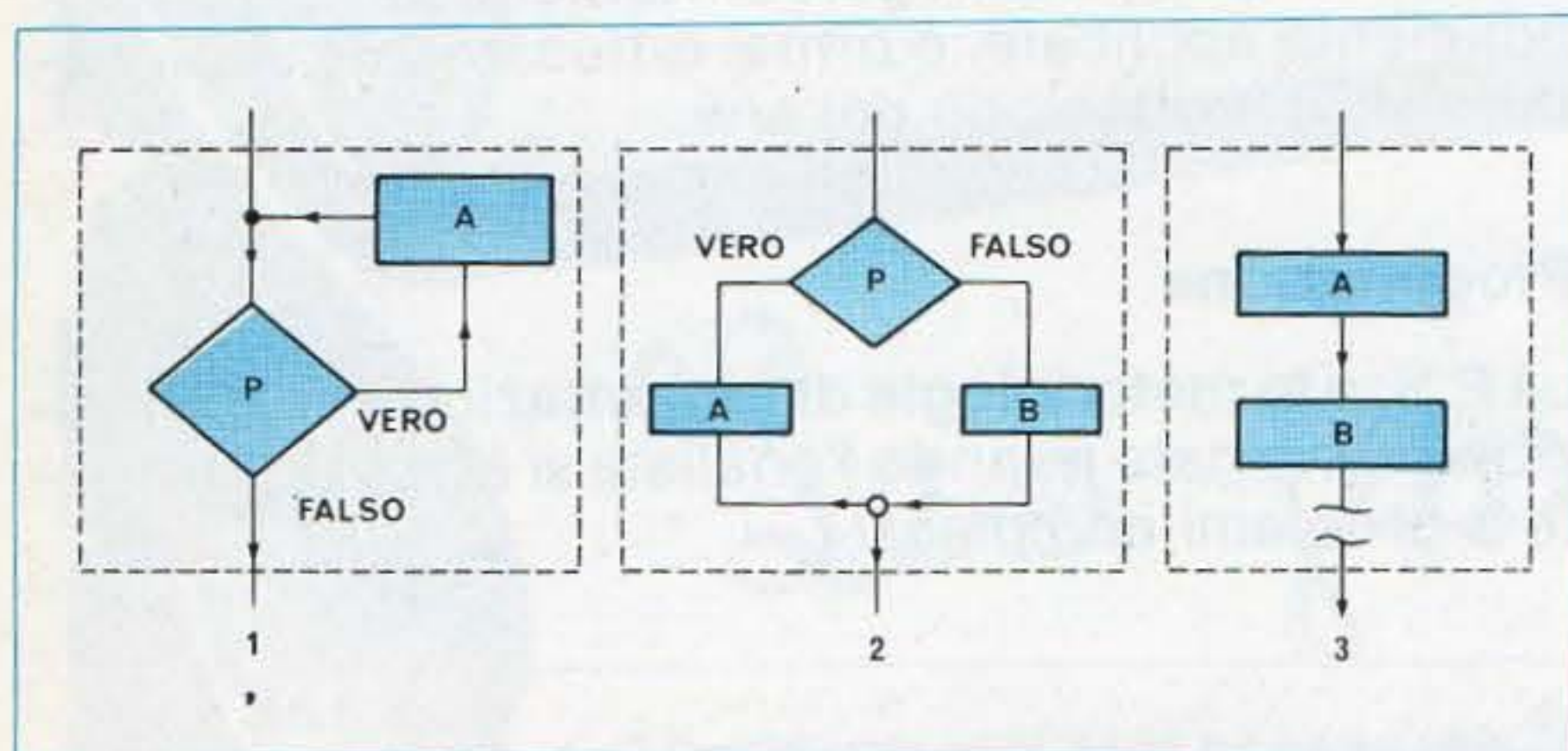
Es: **label** 7,3 [4]

2.-definizione costanti

Possono essere introdotti degli identificatori (sequenze di caratteri alfanumerici il primo dei quali è alfabetico) come sinonimi per dei valori costanti.

Es: **const** PIGRECO = 3.14159

3.-definizione tipi

Ogni identificatore che individua una variabile deve essere associato a un tipo **type**. Questo non solo permette al compilatore di vietare la somma di patate con pomodori come ci insegnavano già in terza elementare ma, obbligando il programmatore a preventivamente definire i dati su cui agirà, facilita la scrittu-

[3] *Non temiamo di svuotare di significato la nozione di complessità né di cadere in un facile psicologismo dicendo che, se per il soggetto che programma c'è un problema, lì c'è complessità.*

[4] *Tutti gli identificatori propri del linguaggio e non altrimenti usabili sono in neretto; essi sono riportati in Tabella 1*

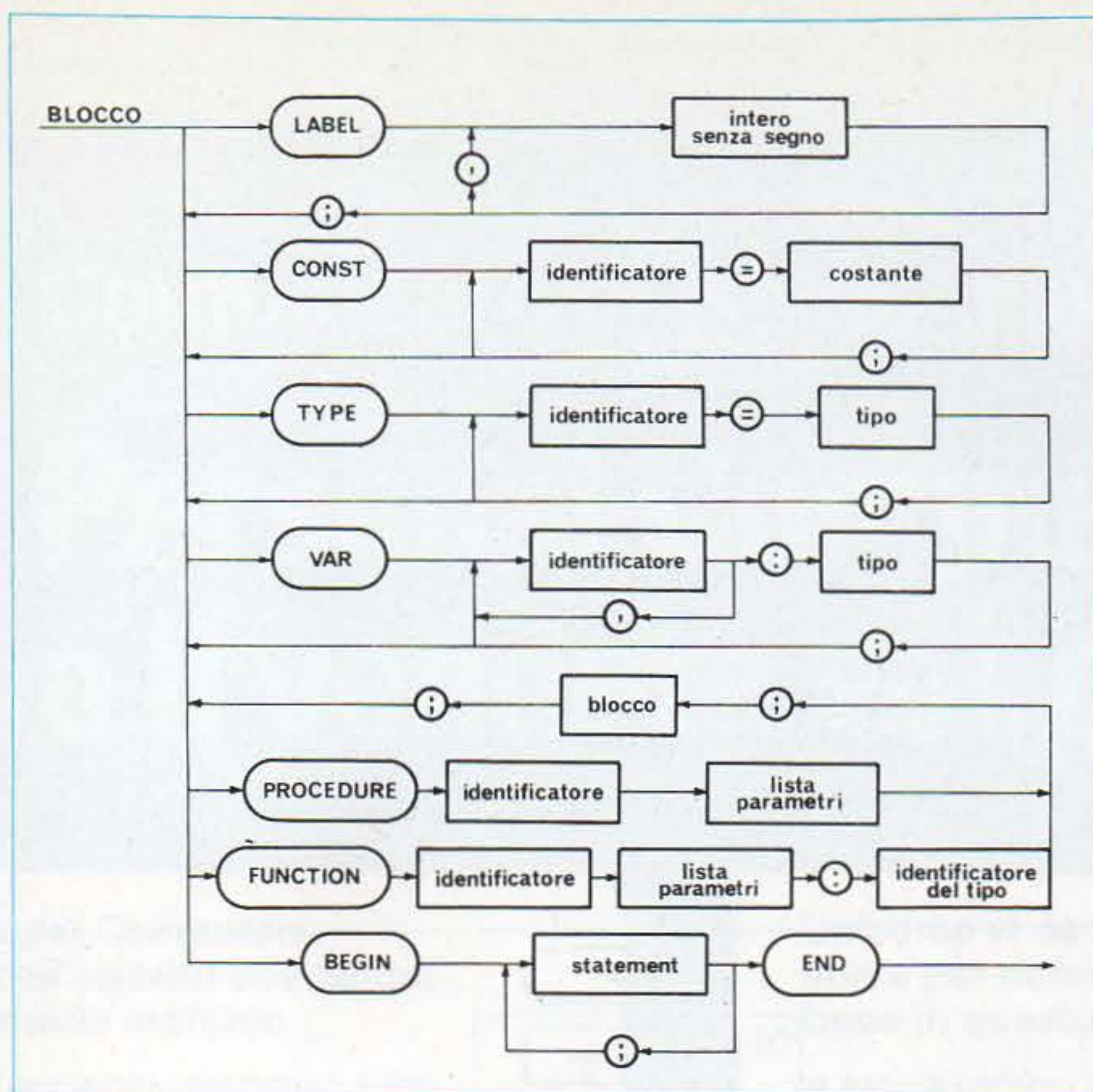



Figura 3 - Diagramma sintattico del "corpo" del programma PASCAL.

ra degli statements ed evita bachi dovuti a dati non chiaramente definiti. In Fig. 4 sono rappresentati i tipi possibili; è da notare come, di contro ad una rigidità delle azioni, PASCAL permette, pur all'interno di regole di costruzioni rigide, delle strutture dati oltremodo flessibili.
Es: **type** SENSO = (UDITO, TATTO, VISTA, ODORATO, GUSTO)

4.-dichiarazioni variabili

Per apparire in uno statement ogni variabile deve essere stata precedentemente definita nella parte di dichiarazione delle variabili: nelle dichiarazioni si associa un identificatore con un tipo.
Es: **var** CONTATORE = **integer**

Nell'esempio si è introdotta una variabile, a cui d'ora innanzi si farà riferimento come *"contatore"*, che può assumere solo valori interi.

5.-dichiarazioni procedure e funzioni

Le procedure / funzioni vanno definite prima del loro uso: esse sono dei sottoprogrammi chiusi che vengono richiamati dal programma principale o da un'altra procedura. Le procedure possono nidificare ed essere recursive. Sull'importanza delle procedure lasciamo di nuovo la parola a Jensen: *"Non bisogna esitare a formulare un'azione come una procedura anche se*

Figura 4 - Albero dei tipi di dati possibili in PASCAL.

+	:	(	and	end	nil	set
−	.	)	array	file	not	then
*	=	[	begin	for	of	to
/	< >	]	case	function	or	type
:=	<	{	const	goto	packed	until
.	<=	}	div	if	procedure	var
.	>=	↑	do	in	program	while
:	>	..	downto	label	record	with
			else	mod	repeat	

Tabella I - Simboli speciali.

chiamata una sola volta, se questo ne aumenta la leggibilità."
La procedura ha formalmente la stessa struttura di un programma; si definiscono locali le variabili conosciute solo a livello della procedura stessa, globali quelle comuni alla procedura ed al programma chiamante: una buona scomposizione sarà quella in cui le variabili globali sono ridotte al minimo.
Es: **procedure** FAQUALCOSA

type **
*
begin
*
*
*
end

6.-statements

Gli statements descrivono le azioni che vengono fatte dall'elaboratore sui dati definiti nelle parti precedenti. Lo statement fondamentale è quello di assegnamento che assegna ad una variabile un nuovo valore; l'operatore di assegnamento è: «:=»
Es: CONTATORE:= **7 div 2**+**8**

Nell'esempio si è posto il valore ottenuto valutando l'espressione a destra nella variabile contatore.
Il simbolo che separa uno statement dall'altro è «;», mentre **begin end** sono le *parentesi* che permettono di raggruppare più statements.
Si considereranno solo due statements oltre l'assegnamento e precisamente lo statement ripetitivo (fig. 2):

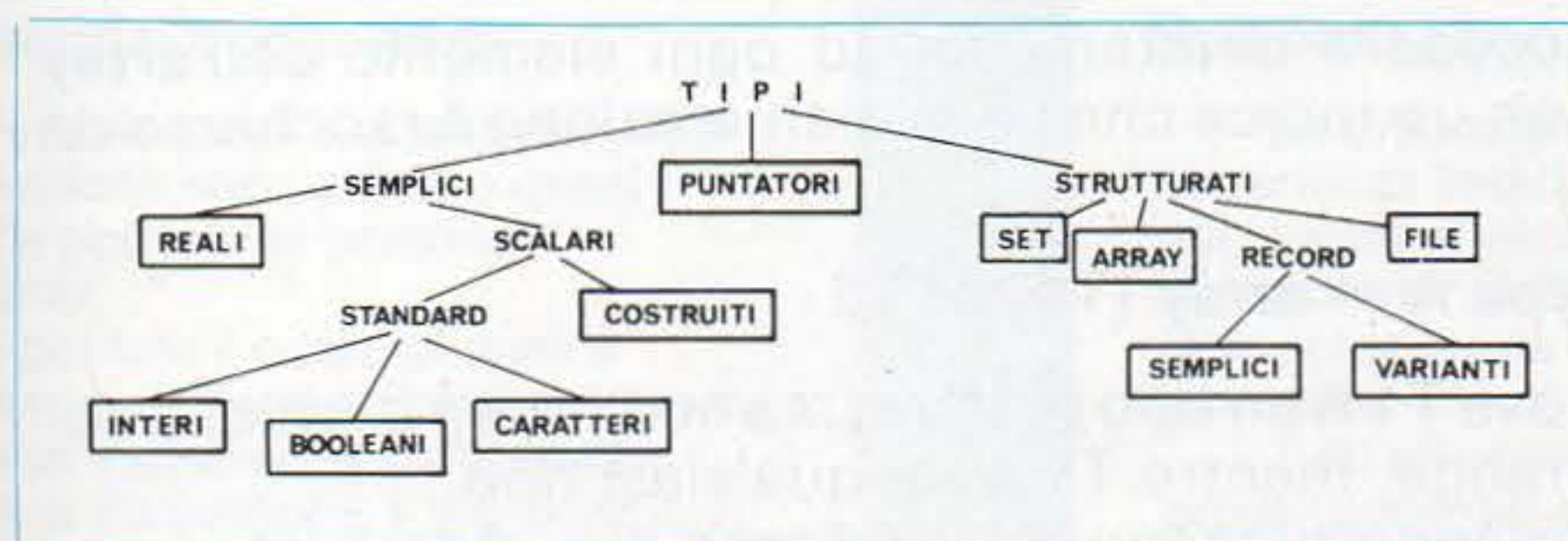
while espressione **do** statement

e lo statement condizionale (fig. 2):

if espressione **then** statement **else** statement

Puntate su	Vincita compresa di posta	Massima puntata
Numero pieno	Puntata × 36	Max
2 Numeri Contigui* (cavallo)	" × 18	Max × 2
4 Numeri Contigui (carré)	" × 9	Max × 4
Sestina	" × 6	Max × 6
Dozzina	" × 3	Max × 12
Rosso/nero	" × 2	Max × 18
Pari/dispari	" × 2	Max × 18
Manque/passe	" × 2	Max × 18

*Contigui va inteso riferito al tavolo roulette: sono ad esempio contigui sia 13 e 16 che 16 e 17. Esempio di CARRE': 26, 27, 29, 30

Tabella II - Descrizioni puntate e banco.

Il problema

Prendendo spunto da un articolo sulla P.S. apparso sul N. 2 di BIT si costruirà un programma di simulazione del gioco della roulette che servirà per analizzare alcuni costrutti del linguaggio.

Specifiche: Il programma deve acquisire le puntate dei giocatori dalla console; generare un numero pseudocasuale compreso tra 0 e 36; pagare le vincite.

Si rimanda alla tabella 2 per i tipi di puntate permesse.

Il listing 1 è uno schema di primo livello che permette un'esplosione delle specifiche. Le scelte a questo livello sono quelle maggiormente vincolanti in quanto condizionano le scomposizioni successive: confrontando con l'esempio del numero precedente di BIT, si vede come là si chiede il nome dei giocatori, qui si conserva l'anonimato (maggior aderenza alla psicologia del giocatore); qui non si danno informazioni sulle regole del gioco, là sì; l'anonimato vieterà certe scelte, come quella di cumulare le vincite e le perdite. A questo primo livello il programma non è che una scomposizione del problema in 5 blocchi logici (procedure): scrivendo in testa delle procedure fittizie si può già provare questa semplice struttura e dare quindi una definizione operativa, tramite l'elaboratore, del problema (listing 2).
Questo approccio, TOP-DOWN radicale, permette:

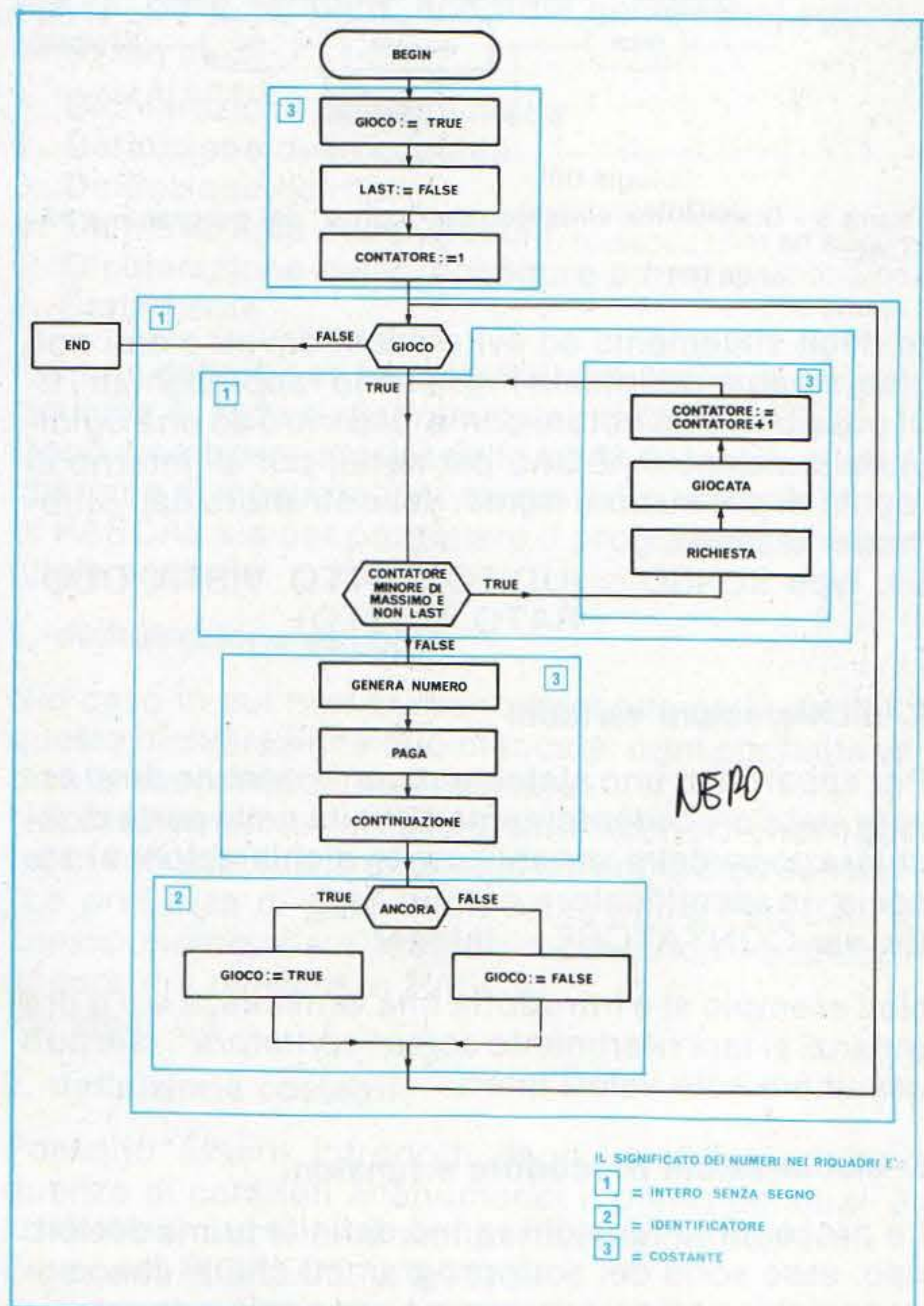
a) di distribuire con buona uniformità il tempo macchina
b) di valutare concretamente l'analisi fin dai livelli alti.
   Nel nostro caso, supposto di usare un video come dispositivo di I/0, si potrebbe valutare se sia più accattivante la visualizzazione del proprio nome o l'anonimato.
c) di mostrare al committente, ad ogni stadio del lavoro, che qualcosa gira; e questo fatto non è da sottovalutare.

Per poter leggere il listing 2 si devono introdurre due procedure standard: READLN e WRITELN.
La prima serve per l'input dei dati da console, la seconda per l'output; si noti che in PASCAL non serve nessun FORMAT associato all'I/0 in quanto il tipo di transcodifica delle variabili di READLN o WRITELN è determinato dal **type**.

I dati

Si analizzano dapprima i dati già introdotti nel listing 2.
Nella parte di identificazione del tipo si è introdotto il tipo STRINGA che è un vettore (**array**) di 80 posizioni costituite da caratteri (**char**).
Il tipo **char** con **boolean, integer, real** fa parte dei tipi scalari standard; **char** è un insieme finito e ordinato di caratteri che dipende dalla periferica di colloquio ed ha come set minimo le lettere latine maiuscole, le cifre 0..9 e lo spazio.
L'indice dell'**array** nell'esempio è rappresentato con due numeri separati da « .. » questa è la rappresentazione dei subranges quali sono dei tipi associati agli scalari (insiemi ordinati) e vengono rappresentati con l'indicazione del minimo e del massimo.
Gli **array** sono un tipo strutturato con un numero fisso di componenti tutti dello stesso tipo; è possibile

```
Begin
| GIOCO := true ; LAST := false
| CONTATORE := 1
| while GIOCO do
|  begin
|  |while (CONTATORE < MASSIMO) and not LAST do
|  | begin
|  | |   RICHIESTA
1  2 3   GIOCATA
|  | |   CONTATORE := CONTATORE + 1
|  | end
|  |GENERA
|  |PAGA
|  |CONTINUAZIONE ?
|  |if ANCORA
|  |  then GIOCO := true
|  |  else GIOCO := false
|  end
end
```

IL SIGNIFICATO DEI NUMERI NEI RIQUADRI È:
1 = INTERO SENZA SEGNO
2 = IDENTIFICATORE
3 = COSTANTE

Listing 1 - Si sono indicati con dei numeri i livelli di "lettura" del programma: si noti come l'uso opportuno dei capoversi (allineamenti dei begin end) visualizza la struttura molto meglio che con il flow corrispondente.

accedere direttamente ad ogni elemento dell'**array** con un indice che nella dichiarazione è racchiuso da «[ ]»:

type A = **array** [T1] **of** T2

dove T1 è un tipo scalare (ma non intero o reale) o subrange, mentre T2 è un qualsiasi tipo.
Volendo proseguire con l'esempio, si tratterà di costruire un pacchetto di informazioni che permetta il computo delle vincite: si può quindi pensare ad un

```
program ROULETTE (INPUT, OUTPUT) input e output sono i
flussi                                    standard di i/o
cost MASSIMO = 20 max giocatori ammessi
type STRINGA = packed array [1. . 80] of char packed istruisce il
compilatore a compattare i car. su un byte
var ST: STRINGA;
GIOCO, ANCORA, LAST: boolean;
CONTATORE: 1. . MASSIMO   sottoinsieme degli interi

procedure RICHIESTA
begin
WRITELN ('RICHIESTA')
end

procedure GIOCATA
begin
WRITELN ('BATTI a SE GIOCHI b se FINE PUNTATE')
READLN (ST)
if ST= B
then LAST := true
end

procedure GENERA
begin
WRITELN ('0. . 36 ROSSO NERO PARI DISPARI MANQUE PASSE. .')
end

procedure PAGA
begin
WRITELN ('GIOCATORE N° 1 10000, . . .')
end

procedure CONTINUAZIONE
begin
WRITELN ('BATTI ancora SE SI CONTINUA')
READLN (ST)
if ST=ANCORA
then ANCORA := true
else ANCORA := false
end

begin  programma principale
*
come listing 1
*
end
```

Listing 2 - Procedure di prova. I commenti sono in lettere minuscole.

Figura 5 - Roulette e banco.

array con tante posizioni quanto il numero massimo dei giocatori e in cui ogni componente è . . . Si potrebbe pensare di memorizzare per ogni giocatore un'immagine del banco delle puntate (fig. 5); come si vede il banco non è omogeneo ma bisogna poter accedere ai suoi campi particolari con dei nomi, in PASCAL esiste un tipo di struttura molto flessibile che può essere usata in casi come questi: **record**.

Il **record** è un tipo strutturato con un numero fisso di componenti ognuno dei quali è accessibile facendo seguire il nome del record da «·» e dal nome del campo.

Nel nostro caso, chiamato GIOCATORE questo tipo, si avrà:

```
type GIOCATORE = record serve per memorizzare
                        le puntate
          NUM:   array [0 . . 36] of integer
          PARI:  integer
          DISP:  integer
          NERO:  integer
          ROSS:  integer
          MANQ:  integer
          PASS:  integer
          PRIM:  integer
          MEDI:  integer
          DERN:  integer
          end
```

In ogni campo si memorizzerà la puntata del giocatore: nel caso in cui è giocata una combinazione di numeri che non ha un proprio campo (per es. cavallo) si dividerà la puntata sui due numeri. La struttura per tutti [5] i giocatori sarà:

```
type TUTTIGIO = array 1..MASSIMO of GIOCATO-
                                       RE
```

Continuando nell'analisi delle strutture dati per il nostro problema, per analogia con il tipo BANCO si definirà un tipo che caratterizza la roulette vera e propria: in effetti l'unica caratteristica che non si può

[5] *Se si pensasse non più ad un gioco didattico ma, come pensabile, al "gioco d'affari" ogni giocatore dovrebbe avere a disposizione un terminale personalizzabile con una scheda rilasciata al giocatore all'ingresso, quando questi versa alla cassa la cifra di cui vuole disporre. Come struttura dati non si può più pensare ad un array di giocatori, ma ci vorrebbe una struttura dinamica con possibilità di inserire e cancellare facilmente: questo si può rendere con liste concatenate che in PASCAL sono gestibili con variabili di tipo pointer:* **type** POINTER = GIOCATORE **var** P: **pointer**

```
program ROULETTE
cost MASSIMO = 20 numero massimo giocatori
type STRINGA = packed array [1. .80] of char;
GIOCATORE = record
                NUM : array [1. .36] of integer;
                PARI, DISP, NERO, ROSSO, MANQ, PASS, PRIM,
                MEDI, DERN: integer
                end;
TUTTIGIO = array [1. .Massimo] of GIOCATORE;
DISPARITA = (PARI, DISPARI, ,) ;
COLORE = (ROSSO, NERO, ,) ;
META = (MANQUE, PASSE, ,) ;
DOZZINE = (PREMIER, MOYEN, DERNIER, ,) ;
NUMBER = record
            DIS: DISPARITA
            COL: COLORE
            MET: META
            DOZ: DOZZINE
            end;
ROULETTE = array [0. .36] of NUMBER

var ST: STRINGA;
GIOCO, ANCORA, LAST: boolean;
CONTATORE: 1. .MASSIMO;
BANCO: TUTTIGIO:
BASE: ROULETTE:
NUMERO: integer;
ZERO: array [1. .36] of 0

procedure COSTRUZIONE chiamata da inizializzazione
                       posiziona colore e disparità dei numeri
begin
for J :=A to B do
   begin
   if ODD (J) funzione standard che verifica disparità
   then begin
        BASE (J). COL := C;
        BASE (J). DIS := DISPARI
        end
else begin
        BASE (J). COL: = D;
        BASE (J). DIS: = PARI
        end
        end
end

procedure colloquio
begin
WRITELN ('PUNTATA :')
READLN (X)
WRITELN ('NUMERO')
READLN ('NUMERO')
end

procedure VERIFICA
begin
CORRETTA := true sempre corretta
end

procedure INIZIA
var A, B: integer; C, D: COLORE; J: integer
begin
A := 1; B := 9; C := ROSSO; D := NERO
COSTRUZIONE
A := 10; B := 18; C := NERO; D := ROSSO;
COSTRUZIONE
A := 19; B := 27; C := ROSSO; D := NERO;
COSTRUZIONE
A := 28; B := 36; C := NERO; D := ROSSO;
COSTRUZIONE
posizionamento 0
BASE (0). COL :=  ;
BASE (0). DIS :=  ;
BASE (0). MET :=  ;
BASE (0). DOZ :=  ;
end

procedure RICHIESTA
begin
WRITELN ('GIOCATORE N°' CONTATORE)
end

procedure GIOCATA
var X: integer; CORRETTA: boolean; NUMERO: 1. .36
begin
COLLOQUIO acquisizione dati e posizionamento e numero
VERIFICA controllo sintattico
if not CORRETTA
then WRITELN ('INPUT NON OK')
else begin
     if X = 0 giocata O: fine puntate
     then LAST := true
     else BANCO (CONTATORE). NUM (NUMERO):= X;
     end
end

function RANDOM: integer
begin
A := A × 27. 182813 + 31. 415917;
A := A - TRUNC (A);
A := TRUNC (A × 50 - 24. 9);
RANDOM := A div 36
end

procedure GENERA
var A : real
begin
A := 1. 23456789; NUMERO = RANDOM
end

procedure PAGA
var X : 1. .MASSIMO; Y: integer
begin
X := 1
while X = CONTATORE do
   begin
   Y := BANCO (X). NUM (NUMERO) × 36
   if Y ≠ 0
   then WRITELN ('GIOCATOR' X 'INCASSA' Y)
   end
end

procedure CONTINUAZIONE
var J : integer
begin
WRITELN ('BATTI «ancora» SE SI CONTINUA')
READLN (ST)
if ST = 'ANCORA'
then begin
     ANCORA := true
     for J := 1 to MASSIMO do
     BANCO (J) := ZERO rateau
     end
else ANCORA := false
end

programma principale
begin
INIZIA
come listing 1
end
```

Listing 3 - Gioco roulette: puntate solo su pieno.

dedurre dal numero 'uscito' è il suo colore; si dovrà creare una tabella di corrispondenza fra numeri e colori. Visto che bisogna etichettare ogni numero con un colore (o nessun colore: lo 0) tanto vale creare dei tipi per ogni 'qualità' dei numeri, costruire quindi un record per ogni numero ed infine un array di record.

```
type  DISPARITA  = (PARI, DISPARI, ,)
type  COLORE     = (ROSSO, NERO, ,)
type  META       = (MANQUE, PASSE, ,)
type  DOZZINE    = (PREMIER, MOYEN, DER-
                    NIER, ,)
```

Il record per ogni numero sarà:

```
type NUMERO = record
                DIS: DISPARITA
                COL: COLORE
                MET: META
                DOZ: DOZZINA
              end
```

mentre il tipo roulette sarà:

```
type  ROULETTE = array 0..36 of NUMERO
```

Avendo introdotta questa struttura dati, bisognerà introdurre una procedura di inizializzazione che metta a punto le 'qualità' di ogni numero (listing 3).
Definita la struttura dati si passa, dopo la dichiarazione delle variabili, alla scomposizione delle procedure

```
var  BANCO: TUTTIGIO; BASE: ROULETTE;
```

Analisi procedure 1° livello

Il prossimo livello sarà quello di permettere un input grezzo: di tutte le possibili puntate si permetterà solo il pieno. Questa scelta è stata fatta perché è intuitivo vedere che la procedura *giocata* è quella che necessita di maggior scomposizione: la specificazione del formato di input è un problema a sua volta complesso che non verrà affrontato.

Si analizzano ora una per una le procedure del listing 1.

richiesta
In questa procedura basterà introdurre la visualizzazione del numero progressivo del giocatore, per cui:

```
procedura RICHIESTA
begin
WRITELN ('GIOCATORE N°' CONTATORE)
end
```

giocata
In questa procedura, a causa della sua complessità, si introdurranno due nuove procedure:
1) COLLOQUIO per acquisizione dati
2) VERIFICA per analisi sintattica
quindi:

```
procedure GIOCATA
var X: real; NUMERO: 0..36; CORRETTA boolean
begin
COLLOQUIO
VERIFICA
if not CORRETTA
then WRITELN ('INPUT NON OK')
else begin
if X = 0 giocata 0 significa fine puntate
then LAST: = true
else BANCO (CONTATORE). NUM (NUMERO):=X
     end
end
```

Le variabili e NUMERO sono posizionate da COLLOQUIO.

genera
Per la generazione del numero casuale si introdurrà una funzione presa da testo di Bowles. Le funzioni sono delle routines che calcolano un singolo valore di tipo scalare o pointer ed il tipo del valore è dichiarato nella intestazione della funzione stessa.

Nel nostro caso:

```
function RANDOM : INTEGER
begin
A        := A×27.182813+31.415917   ;
A        := A−TRUNC (A)              ;
A        := TRUNC (A×50−24.9)        ;
RANDOM := A div 36                    ;
end
```

Nella funzione RANDOM viene usata la funzione standard TRUNC che trasforma un valore reale X nel più grande intero X.

```
procedure GENERA
var A: real
begin
A := 1.23456789
NUMERO := RANDOM
end
```

paga
Questa procedura dovrà scandire BANCO e visualizzare le eventuali vincite:

```
procedure PAGA
var X : 1.. MASSIMO; Y : real
begin
X := 1
while X = CONTATORE do
   begin
   Y:= BANCO (X). NUM (NUMERO) × 36
   if Y = 0
   then WRITELN ('GIOCATORE' X 'INCASSA' Y)
   end
end
```

continuazione
Nella procedura CONTINUAZIONE si deve introdurre, oltre alla richiesta di continuazione del gioco, la 'pulizia del banco'.
Purtroppo avendo eliminati i croupiers sono scomparsi anche i rateaux, per cui ci dovremo limitare a pulire il banco con strutture permesse da PASCAL. Lo statement che permette una risoluzione brillante di questo problema è lo statement **for**:

```
for J:= A to B do STATEMENT
```

dove J è una variabile di controllo, A e B i valori iniziale e finale della stessa. Nel nostro caso, usando come variabile di controllo il numero che individua il giocatore, si avrà:

```
procedure CONTINUAZIONE
varj : INTEGER
begin
WRITELN ('BATTI ancora SE SI CONTINUA')
READLN (ST)
if ST = 'ANCORA'
then begin
     ANCORA := true;
     for J := 1 to MASSIMO do;
     BANCO (CONTATORE) := ZERO
     end
else ANCORA := false
end
```

Dove ZERO è una variabile di tipo GIOCATORE. Il programma relativo a questo livello è riportato nel listing 3.

Il Pascal ed il suo microcomputer

La Western-Digital ha realizzato un microcomputer il cui linguaggio macchina è il P-code generato dal compilatore Pascal dell'Università di San Diego. Questa macchina, MICROENGINE, ha la CPU costituita da 5 circuiti integrati assemblati in contenitori a 40 piedini.
Il primo di essi è un'unità aritmetica, che contiene la decodifica delle microistruzioni e una unità logico-aritmetica.
Il secondo chip è un microsequenziatore che contiene una decodifica delle macroistruzioni (P-CODE), parte della circuiteria di controllo, il contatore delle microistruzioni (P-counter) e la logica di controllo di I/O.
Gli altri tre chips sono ROM 512X22 contenenti il firmware delle microistruzioni e della microdiagnostica.
Insieme questi circuiti integrati formano un processore "stack-oriented" a 16 bit.
L'organizzazione hardware dello "stack" è tale per cui il contenuto dei vari registri viene salvato ogni volta che si entra in una nuova "subroutine".
Alcune delle prestazioni includono la possibilità di indirizzare 128 Kbytes (64 K words) di memoria, quattro livelli di interrupt, e la possibilità di controllare il DMA (direct memory access).
Esiste inoltre una soluzione hardware che consente di gestire il problema della virgola mobile (floating point) in modo da utilizzare questa prestazione, come parte del set di istruzioni.
Tutti i segnali di input/output sono del tipo three-state e sono TTL compatibili. Esiste anche la possibilità di aggiungere altre 2 ROM (512x22) del tipo già descritto, per consentire all'utilizzatore OEM di aggiungere i suoi programmi.
Il sistema così realizzato può coprire la maggioranza delle aree applicative che vanno dai "controllers in real time" ai processori dedicati e in tutte quelle applicazioni che richiedono un linguaggio di programmazione ad alto livello.
La Western Digital Corporation produce il sistema assemblato su una piastra da 8 per 16 pollici con i seguenti blocchi di funzione: 64Kbytes di RAM, 2 porte seriali RS232 con selezione del baud rate da 110 a 19200 baud (bits/secondo), 2 porte parallelo e il floppy disk controller selezionabile per impieghi con dischetti a singola o doppia densità, minifloppy o floppy da 8 pollici. Il floppy disk controller lavora con il DMA e può pilotare fino a quattro drivers dello stesso tipo. Il sistema operativo UCSD-3 viene fornito su floppy disk assieme all'unità.
Il sistema operativo contiene un PASCAL compiler, un BASIC compiler, un file management, un CRT screen oriented editor, un debugging program ed un package per realizzare grafici in modo interattivo. Il MICROENGINE è disponibile anche nella versione "console" per gli utilizzatori finali.
Il PASCAL MICROENGINE ha ora trovato la strada per molti campi applicativi.
E' dunque necessario che chiunque si occupi di computers, minicomputers e microcomputers tenga conto di tale evoluzione per evitare di dover poi puntare ad un obiettivo che altri hanno già raggiunto.
Partire con il MICROENGINE TM significa partire sul piede del futuro.

Andrea Flumiani e Angelo De Biasio
Micolem Data s.r.l.

Analisi procedure 2° livello

Avendo ormai mostrato le strutture fondamentali del linguaggio si tronca il progetto-implementazione a questo livello con un programma le cui funzioni base sono accettabili.

Conclusioni

Questo articolo non voleva essere che uno stimolo alla conoscenza di un linguaggio potente, facile da imparare, molto diffuso e chiaro.
Se l'esposizione ha dato tutt'altra sensazione, la colpa è dell'estensore che, a questo punto, può solamente rimandare il lettore al già citato manuale.
A sua parziale discolpa fa notare che le discontinuità nell'esposizione sono *anche* dovute al tentativo di mostrare il lavoro in fieri, in quanto sarebbe stato mistificante in un articolo *promozionale* su un linguaggio mostrare le strutture pulite nascondendo le fatiche per scrostarle.
Lo scontato finale è preso da una lettera scritta da Blaise Pascal nel 1645 per presentare il suo calcolatore:

"...tu mi sarai grato per il mio impegno nel far si che tutte le operazioni che con gli altri metodi sono faticose, lunghe e poco affidabili diventino facili, semplici, veloci e sicure." ■

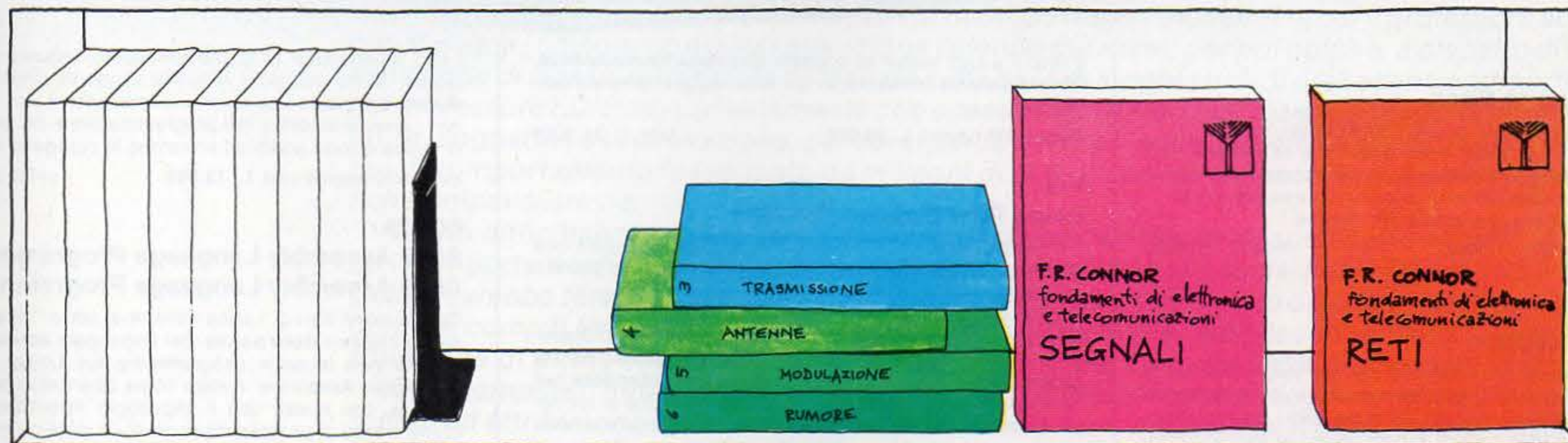
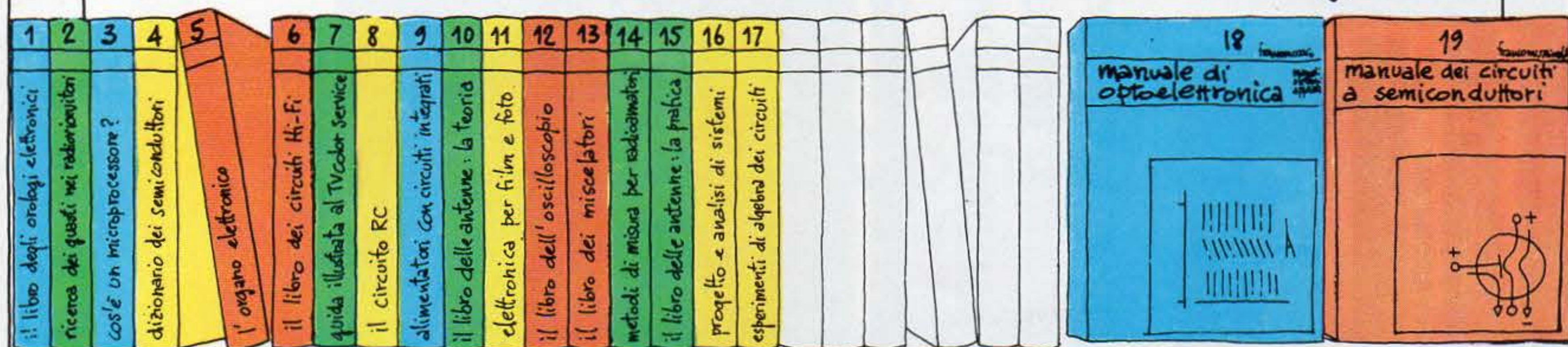
Bibliografia

1) Niklaus Wirth, K. Jensen: Pascal - user manual and report (Springer-Verlag '75)
2) Dahl, Dijkstra, Hoare: Structured programming (Academic press '72)
3) I.E.E.E.: Software design techniques ('76)
4) T.W. Adorno: Minima moralia (Einaudi '54)
5) I.E.E.E.: Structured programming ('75)
6) Ken Bowles: Pascal versus Cobol (Byte n° 8, Agosto '78)
7) Ken Bowles: Problem solving using Pascal (Springer-Verlag '77)
8) Mosconi, D'Urso (a cura di): La soluzione di problemi (Giunti-Barberi '73)
9) Blaise Pascal: La machine aritmetique (da: Oeuvres compètes in NRF Gallimard '54)
10) Yourdon: How to manage structured programming (Yourdon '76)

Due nuovi calcolatori da tavolo Tektronix

Confermando la ferma volontà di primeggiare nel mercato dei calcolatori da tavolo, la Tektronix ha annunciato due nuovi modelli che si affiancano al 4051, modello attualmente in produzione. Il 4052 ed il 4054 offrono una assai maggiore velocità di elaborazione e di trasmissione dati. Il 4054 dispone come standard di uno schermo grafico a 19", mentre il 4051 ed il 4052 utilizzano uno schermo da 11". I tre sistemi hanno caratteristiche adatte ad una molteplicità di impieghi ma si prestano particolarmente ad applicazioni scientifiche quali la ricerca, la progettazione, il design e quelle associate alla rappresentazione del terreno, dove la rappresentazione dei dati in forma grafica offre sostanziali vantaggi all'utilizzatore.
I tre modelli della serie 4050 sono perfettamente trasparenti dal punto di vista del linguaggio di programmazione, della struttura dei dati e dei programmi applicativi, caratteristica unica nel campo dei calcolatori da tavolo. Inoltre il supporto grafico è una dotazione standard di tutti e tre i modelli, come pure la possibilità di ottenere copia su carta dell'immagine sullo schermo.
I due nuovi modelli, 4052 e 4054, utilizzano un rivoluzionario microprocessore, progettato dalla Tektronix, che sfrutta la tecnologia più avanzata a "Bit-slice", grazia alla quale è possibile raggiungere una elevatissima efficienza operativa. La velocità di calcolo di rappresentazione di dati grafici e alfanumerici è superiore a quella di qualunque calcolatore da tavolo oggi in commercio, in termini di tempo valutabili dalle 7 alle 40 volte a seconda della complessità del calcolo.
Il microprocessore ha un set esteso di istruzioni, con aritmetica a virgola mobile, ed una nuova architettura della memoria che contribuisce a rendere così eccezionali le prestazioni del sistema. La serie 4050 è compatibile con una gamma completa di unità periferiche Tektronix, tra cui unità a floppy-disk e da cartuccia magnetica, stampanti, digitalizzatori, videocopiatrici e plotter.
Questi nuovi calcolatori da tavolo saranno presentati in occasione dell'EDP USA 1979 a Milano dal 19 al 22 Giugno.

franco muzzio & c. editore

novità

biblioteca tascabile elettronica

- ☐ 1 L'elettronica e la fotografia, L. 2.400
- ☐ 2 Come si lavora con i transistori, parte prima, L. 2.400
- ☐ 3 Come si costruisce un circuito elettronico, L. 2.400
- ☐ 4 La luce in elettronica, L. 2.400
- ☐ 5 Come si costruisce un ricevitore radio, L. 2.400
- ☐ 6 Come si lavora con i transistori, parte seconda, L. 2.400
- ☐ 7 Strumenti musicali elettronici, L. 2.400
- ☐ 8 Strumenti di misura e di verifica, L. 3.200
- ☐ 9 Sistemi d'allarme, L. 2.400
- ☐ 10 Verifiche e misure elettroniche, L. 3.200
- ☐ 11 Come si costruisce un amplificatore audio, L. 2.400
- ☐ 12 Come si costruisce un tester, L. 2.400
- ☐ 13 Come si lavora con i tiristori, L. 2.400
- ☐ 14 Come si costruisce un telecomando elettronico, L. 2.400
- ☐ 15 Come si usa il calcolatore tascabile, L. 2.400
- ☐ 16 Circuiti dell'elettronica digitale, L. 2.400
- ☐ 17 Come si costruisce un diffusore acustico, L. 2.400
- ☐ 18 Come si costruisce un alimentatore, L. 3.200
- ☐ 19 Come si lavora con i circuiti integrati, L. 2.400
- ☐ 20 Come si costruisce un termometro elettronico, L. 2.400
- ☐ 21 Come si costruisce un mixer, L. 2.400
- ☐ 22 Come si costruisce una radio FM, L. 2.400
- ☐ 23 Effetti sonori per il ferromodellismo, L .2.400

manuali di elettronica applicata

- ☐ 1 Il libro degli orologi elettronici, L. 4.400
- ☐ 2 Ricerca dei guasti nei radioricevitori, L. 4.000
- ☐ 3 Cos'è un microprocessore?, L. 4.000
- ☐ 4 Dizionario dei semiconduttori, L. 4.400
- ☐ 5 L'organo elettronico, L. 4.400
- ☐ 6 Il libro dei circuiti Hi-Fi, L. 4.400
- ☐ 7 Guida illustrata al TVcolor service, L. 4.400
- ☐ 8 Il circuito RC, L. 3.600
- ☐ 9 Alimentatori con circuiti integrati, L. 3.600
- ☐ 10 Il libro delle antenne: la teoria, L. 3.600
- ☐ 11 Elettronica per film e foto, L. 4.400
- ☐ 12 Il libro dell'oscilloscopio, L. 4.400
- ☐ 13 Il libro dei miscelatori, L. 4.800
- ☐ 14 Metodi di misura per radioamatori, L. 4.000
- ☐ 15 Il libro delle antenne: la pratica, L. 3.600
- ☐ 16 Progetto e analisi di sistemi, L. 3.600
- ☐ 17 Esperimenti di algebra dei circuiti, L. 4.800

fondamenti di elettronica e telecomunicazioni

- ☐ 1 Connor - Segnali, L. 3.800
- ☐ 2 Connor - Reti, L. 3.800

novità

- ☐ 18 Ratheiser/Pichler - Manuale di optoelettronica, L. 4.800
- ☐ 19 Benda - Manuale dei circuiti a semiconduttori, L. 4.800
- ☐ 24 Stöckle - Come si lavora con gli amplificatori operazionali, L. 2.400
- ☐ 25 Schierching - Telecomandi a infrarossi per il ferromodellismo, L. 2.400

Prego inviarmi i seguenti volumi. Pagherò in contrassegno l'importo indicato più spese di spedizione.

Tagliando da compilare, ritagliare e spedire in busta chiusa o incollato su cartolina postale a:

SELEZIONE
Via dei Lavoratori, 124
20092 CINISELLO B. (MILANO)

Prego inviarmi i seguenti volumi. Pagherò in contrassegno l'importo indicato più spese di spedizione.

nome ____________________

cognome ____________________

indirizzo ____________________

località ____________________

c.a.p. ____________________

codice fiscale ____________________

LA NOTA

La "formatica", preziosissimo sviluppo applicativo dei Personal Computers

di Aldo Cavalcoli

Come la macchina a vapore ha realizzato una amplificazione della forza muscolare, così il computer amplifica il potere intellettuale.

Partendo da questo assioma, un po' datato, vogliamo sottoporre all'attenzione del lettore una branca di sviluppo, tra le più affascinanti, dei personal computers: la formatica. Questo termine è stato ovviamente coniato dai francesi, come contrazione del noto "informatique", per indicare una nuova scienza in embrione, l'applicazione del computer in campo didattico, inteso in senso lato.

Non si tratta in questo caso di imparare ad usare il computer, ma di usare i computers per imparare, per l'insegnamento, soprattutto riferito alle discipline classiche di tutti i livelli di scolarità.

Il tentativo non è certo nuovo, non mancando illustri esempi di computer-aided-education: la novità consiste nel far uscire il metodo dal ristretto ambito dei laboratori sperimentali e portarlo a contatto con le masse.

Immaginate una scolaresca impegnata nello studio di una disciplina scientifica, diciamo pure la matematica, con un computer per ogni scolaro, e per ciascuno una situazione di interattività con la macchina, tipo domanda e risposta guidata: fantascienza? Folclore? Diciamo piuttosto un futuro molto prossimo.

Innanzitutto i vantaggi. Chi di noi non si ricorda i tragici momenti in cui il professore scorreva con maligna (chiediamo scusa) attenzione l'elenco dei nomi, per poi colpire, a dispetto di tutte le statistiche di classe, duramente compilate sulla base di lunghi studi di precedenti comportamenti del professore? Diciamo onestamente che spesso e volentieri nell'insegnamento si introducono fattori soggettivi e di alienazione, per cui si perde di vista il reale obbiettivo per indulgere a stereotipi di comportamento fossilizzato da millenni di abitudinaria acquiescenza a metodologie didattiche colpevolistiche.

Di ciò difficilmente può essere ritenuto responsabile il docente, anch'egli costretto in necessari comportamenti di verifica del livello di apprendimento della scolaresca. Ed ecco il computer. Cosa dovrebbe fare? Semplicemente restituire al docente il suo ruolo di guida didattica, sgravandolo dei compiti fiscali e di routine, demandando all'allievo la "autogestione" della sua personale situazione di prima verifica di corretta acquisizione dei concetti base di una data materia.

Se il computer, ininterrottamente, "gioca" con l'allievo proponendogli esercizi a difficoltà crescente ed attribuendo alle sue risposte valutazioni anche numeriche, il tutto resta circoscritto all'ambito personale dell'allievo stesso, che in tal modo (così dicono le sperimentazioni in atto) diventa più "audace", posto che intelligentemente verifichi, anche con la sua creatività, il suo rapporto base con la disciplina in oggetto.

Siamo al livello di: cosa fa 4:2? 3! Sbagliato, riprova (ovviamente con tutti i livelli di sofisticazione immaginabile, in funzione della materia e dell'argomento in studio). Al docente resterebbe il ruolo fondamentale di verifica dopo la sgrossatura, di definizione delle linee didattiche, di studio e di ricerca, come sarebbe giusto ed auspicabile.

Se poi è vero, come è vero, che il computer, a basso costo, sta per invadere tutti i settori della vita umana, è inevitabile che uno studente si "abitui" al computer, come del resto è sacrosanto che si abitui il docente, sia esso proveniente dall'università che dagli istituti tecnici od altri gradi di scuole (asilo? Perché no!).

Ma non vi immaginate le infinite possibilità educative dei personal computer? Come esperienza professionale abbiamo avuto modo di dialogare a lungo con docenti di istituti tecnici nell'ambito di una serie di corsi sul microprocessore Z-80 da noi realizzati, e sempre abbiamo riscontrato un notevole entusiasmo, quasi mai supportato però da una chiara visione della linea di sviluppo dell'argomento microcomputer e personal computer, con riferimento alla loro sfera professionale.

La formatica è ancora lontana, mancano investimenti, mancano volontà: in una parola manca una chiara conoscenza dell'argomento.

Ancora, la stessa metodologia è istintivamente applicabile in campo industriale, dove ad esempio si può demandare ad un computer il compito di educare, in via preliminare, i neo assunti ai futuri compiti che li aspettano in azienda, senza per questo distogliere personale specializzato da momenti produttivi.

Sarebbe confortante, e senz'altro lusinghiero, suscitare con questa nota qualche interesse diffuso, al limite un inizio di dibattito sull'argomento "formatica": questo è un aperto invito della rivista BIT a quanti si sentono sensibili alle implicazioni didattiche dei personal computers.

PERSONAL COMPUTER

Giochiamo a Dama con il computer!

Tratto da *"Basic computer games"*. Creative computing Press.
Distribuito in Italia dalla Jackson Editrice s.r.l.

Con questo programma si gioca a dama. Le pedine mosse dal calcolatore sono contrassegnate da una "X", le vostre da uno "0". Per eseguire una mossa, si devono specificare le coordinate (X,Y) della pedina che deve essere spostata.
La posizione (0,0) si trova in basso a sinistra; la coordinata "X" specifica la distanza a destra della posizione (0,0) (cioè la colonna) e la coordinata "Y" specifica la distanza sopra la posizione (0,0) (cioè la riga).
Occorre anche specificare dove si vuol muovere la pedina.
Benché il programma non sia così raffinato da saper comunicare quando la partita è vinta, ad ogni modo si può giocare una partita divertente.
Il programma è eseguibile usando il MITS Altair 8K BASIC, revisione 4.0.

Esecuzione del programma

```
DAMA

QUESTO E' IL GIOCO DELLA DAMA.
IL COMPUTER E' X, E VOI SIETE 0. IL
COMPUTER MUOVE PER PRIMO.
LA SCACCHIERA E' RIFERITA A UN SI-
STEMA DI COORDINATE, (0,0) E' LA CA-
SELLA NELL'ANGOLO IN BASSO A SI-
NISTRA (0,7) E' LA CASELLA NELLO
ANGOLO IN BASSO A SINISTRA (0,7) E'
LA CASELLA NELL'ANGOLO IN ALTO A
SINISTRA (7,0) E' LA CASELLA NELLO
ANGOLO IN BASSO A DESTRA (7,7) E'
LA CASELLA NELL'ANGOLO IN ALTO A
DESTRA IL COMPUTER STAMPERA" +
TO' QUANDO POTETE FARE UN'ALTRA
MANGIATA. STAMPA DUE NUMERI NE-
GATIVI SE NON POTETE PIU' MANGIA-
RE.
```

```
DA 1 5 A 0 4

. X . X . X . X
X . X . X . X .
. . . X . X . X
X . . . . . . .
. . . . . . . .
0 . 0 . 0 . 0 .
. 0 . 0 . 0 . 0
0 . 0 . 0 . 0 .
```

```
DA? 2,2
A? 3,3
DA 0 6 1 1 5

. X . X . X . X
. . X . X . X .
. X . X . X . X
X . . . . . . .
. . . 0 . . . .
0 . . . 0 . 0 .
. 0 . 0 . 0 . 0
0 . 0 . 0 . 0 .
```

```
DA? 0,2
A? 1,3
DA 0 4 A 2 2

. X . X . X . X
. . X . X . X .
. X . X . X . X
. . . . . . . .
. . . 0 . . . .
. . X . 0 . 0 .
. 0 . 0 . 0 . 0
0 . 0 . 0 . 0 .
```

```
DA? 3,1
A? 1,3
+? -1,-1
DA 1 5 1 0 1

. X . X . X . X
. . X . X . X .
. . . X . X . X
X . . . . . . .
. 0 . 0 . . . .
. . . . 0 . 0 .
. 0 . . . 0 . 0
0 . 0 . 0 . 0 .
```

```
DA? 4,0
A? 3,1
DA 0 4 A 2 2 A 4 0

. X . X . X . X
. . X . X . X .
. . . X . X . X
. . . . . . . .
. . . 0 . . . .
. . . . 0 . 0 .
. 0 . . . 0 . 0
0 . 0 . X* . 0 .
```

```
DA? 6,2
A? 5,3
DA 4 0 A 6 2 A 4 4 A 2 2   2 2

. X . X . X . X
. . X . X . X .
. . . X . X . X
. . . . . . . .
. . . . . . . .
. . X* . 0 . . .
. 0 . . . . . 0
0 . 0 . . . 0 .
```

```
DA? 1,1
A? 3,3
+ TO?-1, -1
DA 2 6 A 1 5

. X . X . X . X
. . . . X . X .
. X . X . X . X
. . . . . . . .
. . . 0 . . . .
. . . . 0 . . .
. . . . . . . 0
0 . 0 . . . 0 .
```

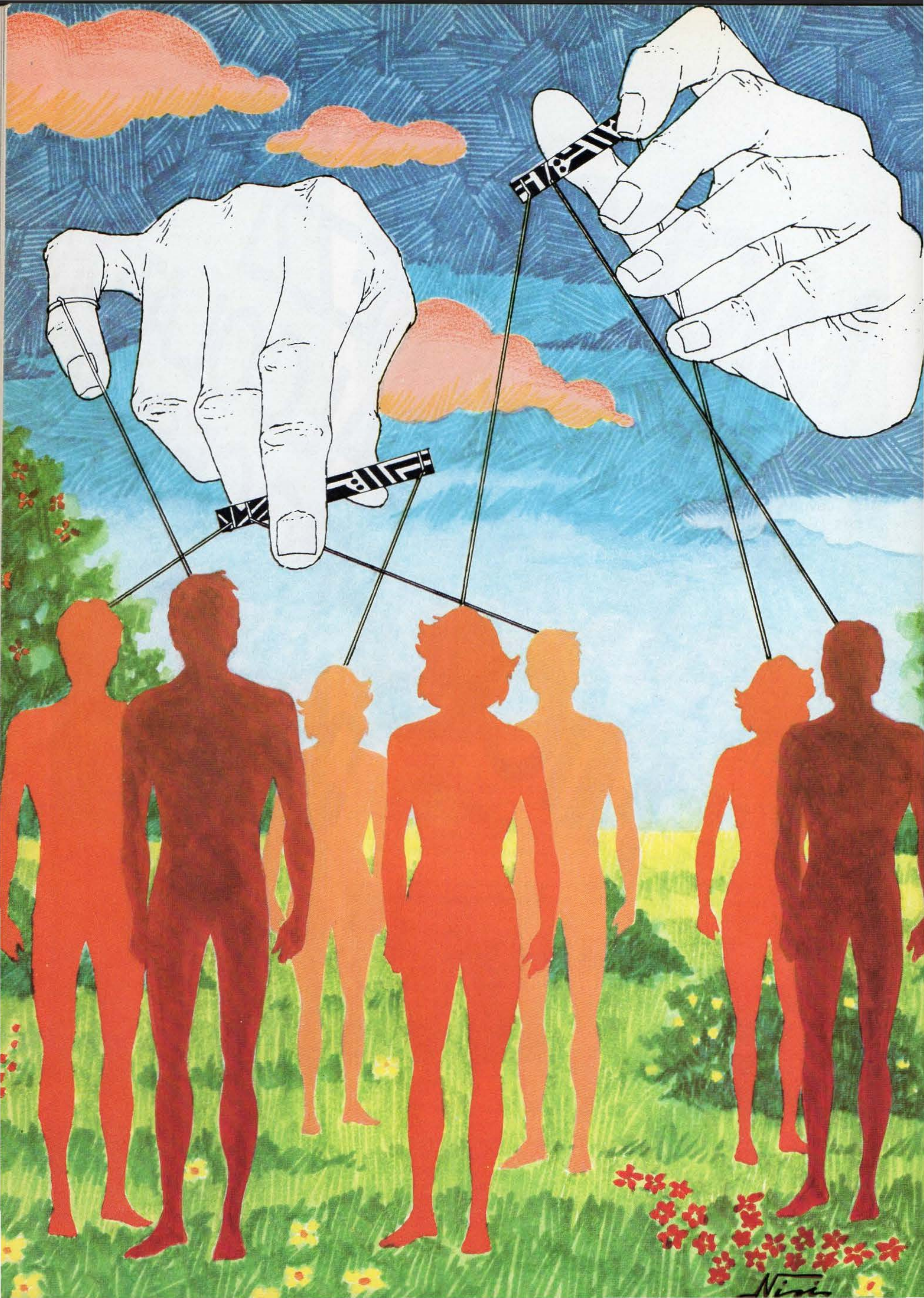
```
DA? 0,0
A? 1,1
DA 1 5 A 0 4

.   X   .   X   .   X   .   X
.   .   .   .   X   .   X   .
.   .   .   X   .   X   .   X
X   .   .   .   .   .   .   .
.   .   .   0   .   .   .   .
.   .   .   .   0   .   .   .
.   0   .   .   .   .   .   0
.   .   0   .   .   .   0   .

DA? 1,1
A? 2,2
DA 3 5 A 4 4

.   0   .   .   .   0   .   .
0   .   .   .   .   .   .   .
.   .   .   0   .   0   .   .
.   .   .   .   0   .   .   .
.   .   .   X   .   .   .   X
X   .   X   .   .   .   .   .
.   X   .   X   .   .   .   .
X   .   X   .   X   .   X   .

DA? 3,3
A? 2,4
DA 5 5 A 6 4

.   X   .   X   .   X   .   X
.   .   .   .   X   .   X   .
.   .   .   .   .   .   .   X
X   .   0   .   X   .   X   .
.   .   .   .   .   .   .   .
.   .   0   .   0   .   .   .
.   .   .   .   .   .   .   0
.   .   0   .   .   .   0   .

DA? 0,2
DA? 2,0
A 3,1
DA 4 6 A 5 5

.   X   .   X   .   X   .   X
.   .   .   .   .   .   X   .
.   .   .   .   .   X   .   X
X   .   0   .   X   .   X   .
.   .   .   .   .   .   .   .
.   .   0   .   0   .   .   .
.   .   .   0   .   .   .   0
.   .   .   .   .   .   0   .
```

```
5 PRINT TAB(32);"DAMA"
15 PRINT:PRINT:PRINT
20 PRINT "QUESTO E' IL GIOCO DELLA DAMA. IL COMPUTER E' X,"
25 PRINT "E VOI SIETE O. IL COMPUTER MUOVE PER PRIMO."
30 PRINT "LA SCHACCHIERA E' RIFERITA A UN SISTEMA DI COORDINATE."
35 PRINT "(0,0) E' LA CASELLA NELL'ANGOLO IN BASSO A SINISTRA"
40 PRINT "(0,7) E' LA CASELLA NELL'ANGOLO IN ALTO A SINISTRA"
45 PRINT "(7,0) E' LA CASELLA NELL'ANGOLO IN BASSO A DESTRA"
50 PRINT "(7,7) E' LA CASELLA NELL'ANGOLO IN ALTO A DESTRA"
55 PRINT "IL COMPUTER STAMPERA' '+TO' QUANDO POTETE FARE UN'ALTRA"
60 PRINT "MANGIATA. STAMPA DUE NUMERI NEGATIVI SE NON POTETE"
62 PRINT "PIU' MANGIARE."
65 PRINT:PRINT:PRINT
80 DIM R(4),S(7,7):G=-1:R(0)=-99
90 DATA 1,0,1,0,0,0,-1,0,0,1,0,0,0,-1,0,-1,15
120 FORX=0TO7:FORY=0TO7:READJ:IFJ=15THEN180
160 S(X,Y)=J:GOTO 200
180 RESTORE:READS(X,Y)
200 NEXTY,X
230 FORX=0TO7:FORY=0TO7:IFS(X,Y)>-1THEN350
310 IFS(X,Y)=-1THENFORA=-1TO1STEP2:B=G:GOSUB650:NEXTA
330 IFS(X,Y)=-2THENFORA=-1TO1STEP2:FORB=-1TO1STEP2:GOSUB650:NEXTB,A
350 NEXTY,X:GOTO1140
650 U=X+A:V=Y+B:IFU<0ORU>7ORV<0ORV>7THEN870
740 IFS(U,V)=0THENGOSUB910:GOTO870
770 IFS(U,V)<0THEN870
790 U=U+A:V=V+B:IFU<0ORV<0ORU>7ORV>7THEN870
850 IFS(U,V)=0THENGOSUB910
870 RETURN
910 IFV=0ANDS(X,Y)=-1THENQ=Q+2
920 IFABS(Y-V)=2THENQ=Q+5
960 IFY=7THENQ=Q-2
980 IFY=0ORU=7THENQ=Q+1
1030 FORC=-1TO1STEP2:IFU+C<0ORU+C>7ORV+G<0THEN1080
1035 IFS(U+C,V+G)<0THENQ=Q+1:GOTO1080
1040 IFU-C<0ORU-C>7ORV-8>7THEN1080
1045 IFS(U+C,V+G)>0AND(S(U-C,V-G)=0OR(U-C=XANDV-G=Y))THENQ=Q-2
1080 NEXTC:IFQ>R(0)THENR(0)=Q:R(1)=X:R(2)=Y:R(3)=U:R(4)=V
1100 Q=0:RETURN
1140 IFR(0)=-99THEN1880
1230 PRINTCHR$(30)"DA"R(1);R(2)"A"R(3);R(4);:R(0)=-99
1240 IFR(4)=0THENS(R(3),R(4))=-2:GOTO1310
1230 S(R(3),R(4))=S(R(1),R(2))
1310 S(R(1),R(2))=0:IFABS(R(1)-R(3))<>2THEN1420
1330 S((R(1)+R(3))/2,(R(2)+R(4))/2)=0
1340 X=R(3):Y=R(4):IFS(X,Y)=-1THENB=-2:FORA=-2TO2STEP4:GOSUB1370
1350 IFS(X,Y)=-2THENFORA=-2TO2STEP4:FORB=-2TO2STEP4:GOSUB1370:NEXTB
1360 NEXTA:IFR(0)<>-99THENPRINT"A"R(3);R(4);:R(0)=-99:GOTO1240
1365 GOTO1420
1370 U=X+A:V=Y+B:IFU<0ORU>7ORV<0ORV>7THEN1400
1380 IFS(U,V)=0ANDS(X+A/2,Y+B/2)>0THENGOSUB910
1400 RETURN
1420 PRINT:PRINT:PRINT:FORY=7TOSTEP-1:FORX=0TO7:I=5*X:PRINTTAB(I);
1430 IFS(X,Y)=0THENPRINT".";
1470 IFS(X,Y)=1THENPRINT"0";
1490 IFS(X,Y)=-THENPRINT"X";
1510 IFS(X,Y)=-2THENPRINT"X*";
1530 IFS(X,Y)=2THENPRINT"0*";
1550 NEXTX:PRINT" ":PRINT:NEXTY:PRINT
1590 INPUT "DA";E,H:X=E:Y=H:IFS(X,Y)<=0THEN 1590
1670 INPUT "A";A,B:X=A:Y=B
1680 IFS(X,Y)=0ANDABS(A-E)<=2ANDABS(A-E)=ABS(B-H)THEN1700
1690 PRINTCHR$(7)CHR$(11);:GOTO1670
1700 I=46
1750 S(A,B)=S(E,H):S(E,H)=0:IFABS(E-A)<>2THEN1810
1800 S((E+A)/2,(H+B)/2)=0
1802 INPUT "+TO";A1,B1:IF A1<0 THEN 1810
1804 IFS(A1,B1)<>0ORABS(A1-A)<>2ORABS(B1-B)<>2THEN1802
1806 E=A:H=B:A=A1:B=B1:I=I+15:GOTO1750
1810 IFB=7THENS(A,B)=2
1830 GOTO230
1880 PRINT "AVETE VINTO"
1890 END
```

La COMPREL, distributrice della gamma completa dei prodotti Western Digital, fra i quali sono da segnalare la linea USART e la linea Floppy-Disk Controller per drives di diskette a singola/doppia faccia, singola/doppia densità, comunica la distribuzione dei microcomputer PASCAL MICRO-ENGINE della Microengine Company, consociata della Western Digital. gital.

Questo sistema è un desk top microcomputer sviluppato attorno a 5 chip LSI/MOS della Western Digital, i quali implementano un processore a 16 bit stack-oriented che esegue direttamente il P-code Pascal della Università di San Diego, California (USDC).

Le caratteristiche hardware del sistema sono:
- processore WD/9000,
- 64 Kbytes di RAM,
- due porte seriali asincrone RS-232 con baud-rate selezionabile da switches,
- due porte parallele,
- FDC per il controllo fino a 4 drives floppy-disk che possono essere a singola/doppia faccia (la scelta è fatta tramite switches),
- alimentatore.

Il software fornito consiste nel sistema operativo Pascal UCDS, che comprende
- compilatore Pascal,
- compilatore BASIC,
- file manager,
- editor screen-oriented,
- debug.

Life
(Il gioco della vita)

Questo gioco fu descritto per la prima volta in un articolo di Martin Gardiner nel numero di Ottobre 1970 della rivista Scientific American.
Il gioco, propriamente, fu inventato da John Conway del Gonville and Caius College dell'università di Cambridge, Inghilterra.
Nel gioco classico, gli individui sono indicati per mezzo di simboli (puntini, asterischi...) sopra un grande reticolato e muoiono, si riproducono in accordo ad alcune semplici regole genetiche.
I criteri adottati da Conway per scegliere le leggi genetiche sono:

1) Non ci deve essere alcuna disposizione iniziale dalla quale risulti evidente che la popolazione possa crescere senza limite.
2) Ci devono essere disposizioni iniziali che apparentemente possono crescere senza limite.
3) Ci devono essere semplici disposizioni iniziali che crescono e mutano per un certo periodo di tempo prima di arrivare ad una fine in tre modi possibili:
   - scomparire completamente (per sovrappopolamento o isolamento);
   - venirsi a trovare in una stabile configurazione che rimanga in seguito inalterata;
   - giungere ad un punto in cui la stessa situazione si ripete ciclicamente dopo due o più variazioni.

In breve, le regole devono essere tali da rendere relativamente imprevedibile il comportamento della popolazione.
Le leggi genetiche di Conway sono semplici. Innanzitutto si deve notare che ogni cella del reticolato (supposto come un piano infinito) ha otto celle contigue, quattro adiacenti ortogonalmente e quattro adiacenti in diagonale.

Tratto da *"Basic computer games"* Creative Computing Press
Distribuito in Italia dalla Jackson Editrice s.r.l.

Le regole sono le seguenti:

1) Sopravvivenza. Ogni simbolo con due o tre simboli contigui sopravvive nella generazione successiva.
2) Morte. Ogni simbolo con quattro o più simboli contigui muore (viene soppresso) a causa di sovrappopolamento. Ogni simbolo con uno o nessun simbolo vicino muore per isolamento.
3) Nascita. Ogni cella vuota *adiacente esattamente a tre simboli* è una cella generatrice. Un simbolo la occuperà alla mossa successiva.

È importante tener presente che tutte le nascite e le morti avvengono contemporaneamente, globalmente esse costituiscono un'unica generazione.

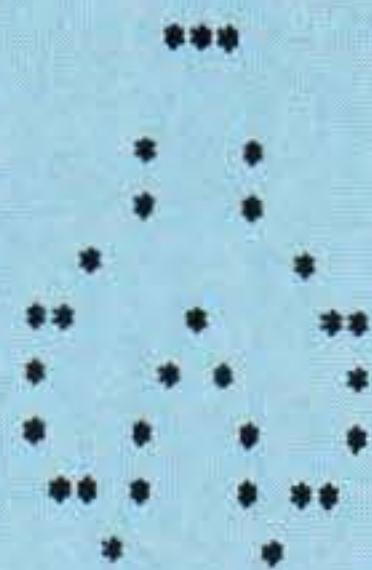
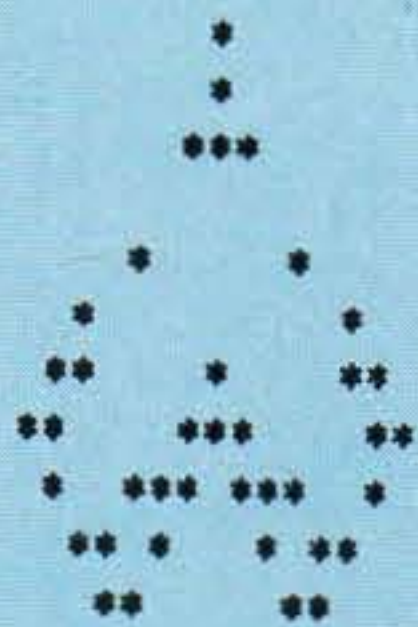
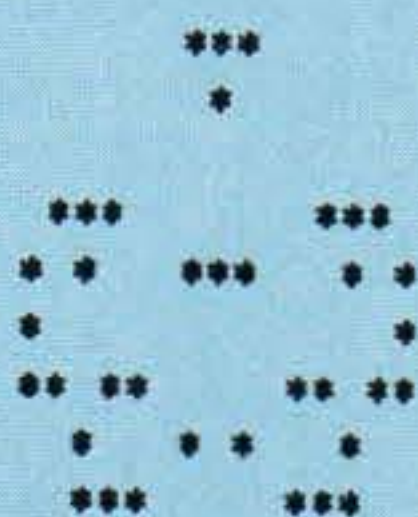
Noterete che la popolazione subisce costantemente cambiamenti insoliti, a volte positivi e sempre insospettati. In qualche caso la società alla fine muore (tutti i simboli scompaiono), sebbene ciò non possa capitare se non dopo molte generazioni.

La maggior parte delle configurazioni di partenza raggiungono situazioni stabili - Conway le chiama "nature morte" - che non possono cambiare. O configurazioni che si ripetono oscillando all'infinito.

Le configurazioni che inizialmente non sono simmetriche tendono a diventarlo. Una volta che questo è accaduto, la simmetria non si perde più anche se può aumentare di dimensione.
Conway usò un DEC PDP-7 con un dispositivo grafico per osservare le popolazioni a lunga vita. Potrà essere più apprezzabile osservarle su un CRT che su un terminale hard-copy.
Poichè il MITS 8K BASIC non ha l'istruzione LINE INPUT, per inserire degli spazi nella disposizione, dovete battere "." all'inizio della riga. Questo sarà trasformato in uno spazio dal BASIC, permettendo in tal modo di stampare degli spazi. Battendo DONE indicate che avete terminato di inserire la disposizione.

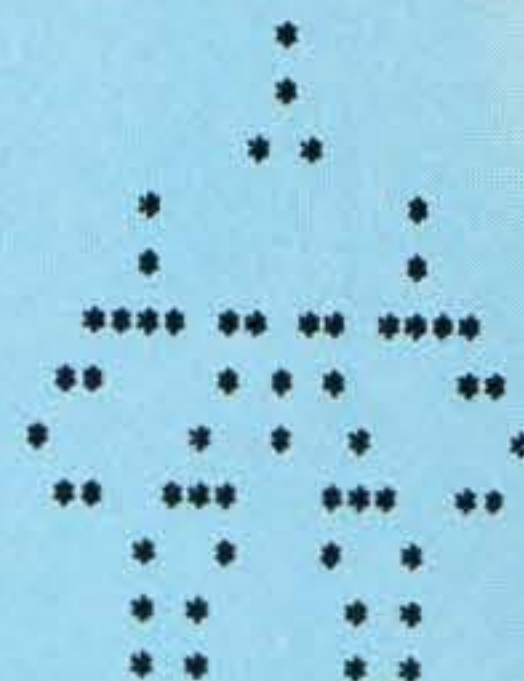
```
2 PRINT TAB(34);"VITA"
6 PRINT: PRINT: PRINT
8 PRINT "INTRODUCI LA TUA DISPOSIZIONE"
9 X1=1: Y1=1: X2=24: Y2=70
10 DIM A(24,70),B$(24)
20 C=1
30 INPUT B$(C)
40 IF B$(C)="FATTO" THEN B$(C)="": GOTO 80
50 IF LEFT$(B$(C),1)="." THEN B$(C)=" "+RIGHT(B$(C),LEN(B$( ))-1)
60 C=C+1
70 GOTO 30
80 C=C-1: L=0
90 FOR X=1 TO C-1
100 IF LEN(B$(X))>  THEN L=LEN(B$(X))
110 NEXT X
120 X1=11-C/2
130 Y1=33-L/2
140 FOR X=1 TO C
150 FOR Y=1 TO LEN(B$(X))
160 IF MID$(B$(X),Y,1)<>" " THEN A( 1+X,Y1+Y)=1:P=P+1
170 NEXT Y
180 NEXT X
200 PRINT:PRINT:PRINT
210 PRINT"GENERAZIONE:";G,"POPOLAZIONE:";P;:IFI9THEN PRINT"NON VALIDO";
215 X3=24: Y3=70: X4=1: Y4=1: P=0
220 G=G+1
225 FOR X=1 TO X1-1:PRINT: NEXT X
230 FOR X=X1 TO X2
240 PRINT
250 FOR Y=Y1 TO Y2
253 IF A(X,Y)=2 THEN A(X,Y)=0: GOTO 270
256 IF A(X,Y)=3 THEN A(X,Y)=1: GOTO 261
260 IF A(X.Y)<>1 THEN 270
261 PRINT TAB(Y);"*";
262 IF X<X3 THEN X3=X
264 IF X>X4 THEN X4=X
266 IF Y<Y3 THEN Y3=Y
268 IF Y>Y4 THEN Y4=Y
270 NEXT Y
290 NEXT X
295 FOR X=X2+1 TO 24: PRINT: NEXT X
299 X1=X3: X2=X4: Y1=Y3: Y2=Y4
301 IF X1<3 THEN X1=3: I9=-1
303 IF X2>22 THEN X2=22: I9=-1
305 IF Y1<3 THEN Y1=3: I9=-1
307 IF Y2>68 THEN Y2=68: I9=-1
309 P=0
500 FOR X=X1-1 TO X2+1
510 FOR Y=Y1-1 TO Y2+1
520 C=0
530 FOR I=X-1 TO X+1
540 FOR J=Y-1 TO Y+1
550 IF A(I,J)=1 OR A(I,J)=2 THEN C=C+1
560 NEXT J
570 NEXT I
580 IF A(X,Y)=0 THEN 610
590 IF C<3 OR C>4 THEN A(X,Y)=2: GOTO 600
595 P=P+1
600 GOTO 620
610 IF C=3 THEN A(X,Y)=3: P=P+1
620 NEXT Y
630 NEXT X
635 X1=X1-1: Y1=Y1-1: X2=X2+1: Y2=Y2+1
640 GOTO 210
650 END
```

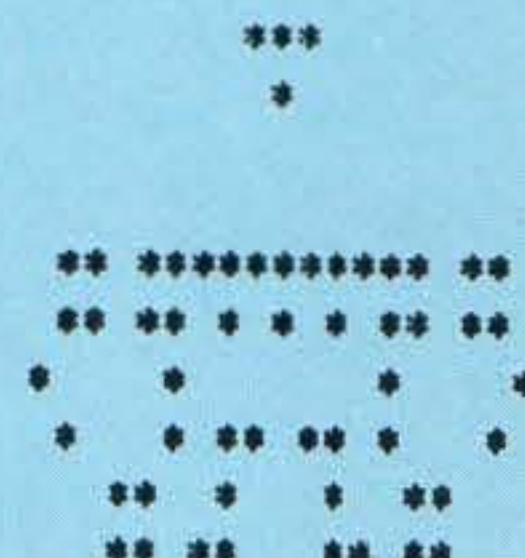
```
                     LIFE
 CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

INTRODUCI LA TUA DISPOSIZIONE:
? .  ***
? . *  *
? .*   *
? FATTO


GENERAZIONE: 0          POPOLAZIONE: 7



                    ***
                    * *
                   *   *

GENERAZIONE: 1          POPOLAZIONE: 8



                     *
                    ***
                   ** **

GENERAZIONE: 2          POPOLAZIONE: 9



                    ***
                   *   *
                   ** **

GENERAZIONE: 3          POPOLAZIONE: 10



                     *
                    ***
                   *   *
                   ** **

GENERAZIONE: 4          POPOLAZIONE: 12



                    ***
                    ***
                   *   *
                   ** **
```

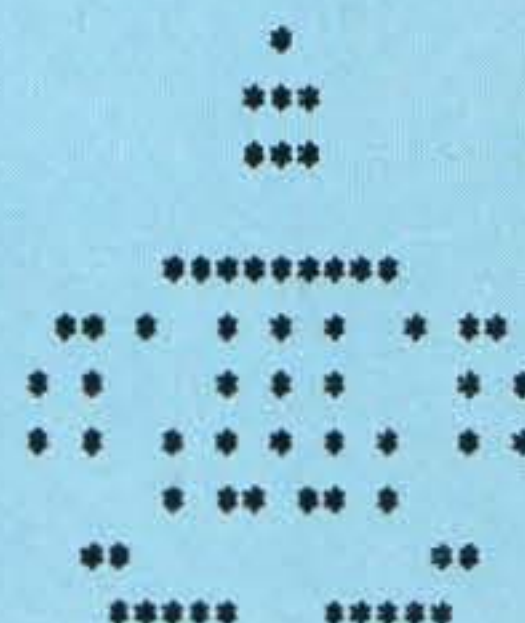
GENERAZIONE: 13 POPOLAZIONE: 30

GENERAZIONE: 14 POPOLAZIONE: 39

GENERAZIONE: 15 POPOLAZIONE: 37

GENERAZIONE: 16 POPOLAZIONE: 48

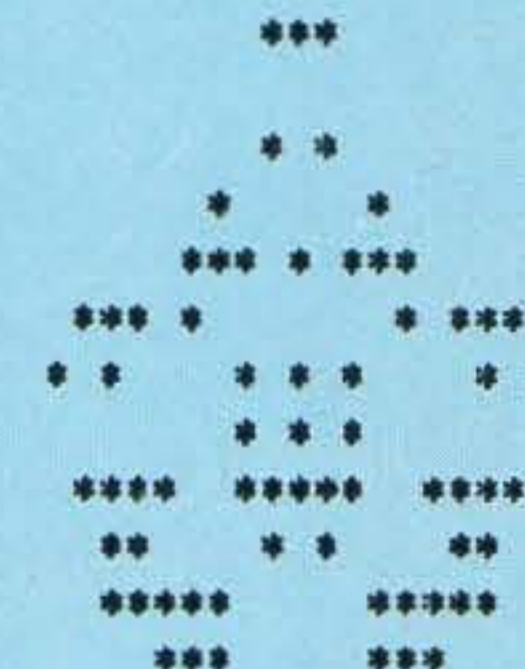
GENERAZIONE: 17 POPOLAZIONE: 46

GENERAZIONE: 18 POPOLAZIONE: 54

```
         *
         *
        * *
    *         *
    *         *
  **** ** ** ****
 **    * * *    **
*     *  *  *     *
 **  ***   ***  **
    *  *   *  *
    * *     * *
    * *     * *
```

GENERAZIONE: 19 POPOLAZIONE: 56

```
        ***
         *


 ** *********** **
 ** ** * * * ** **
*    *       *    *
 *   * ** ** *   *
   **  *   *  **
   ** **   ** **
```

GENERAZIONE: 20 POPOLAZIONE: 61

```
         *
        ***
        ***

     *********
 ** *  * * *  * **
* *    * * *    * *
* *  * * * * *  * *
     * ** ** *
  **           **
   *****   *****
```

GENERAZIONE: 21 POPOLAZIONE: 67

```
        ***

        * *
      *     *
     *** * ***
 *** *       * ***
* *    * * *    * *
       * * *
 ****  *****  ****
  **    * *    **
  *****     *****
    ***     ***
```

Il Nanocomputer® NBZ80

di **Aldo Cavalcoli** - MIPRO s.r.l.

Il Nanocomputer®NBZ80 della SGS-ATES è uno tra i più validi strumenti didattici attualmente sul mercato per l'insegnamento dell'uso dei sistemi a microprocessore. Questo articolo intende presentare le caratteristiche base di questo sistema.

Elementi qualificanti di un sistema didattico

Attualmente sul mercato sono presenti molti sistemi, definiti didattici da parte dei costruttori, venditori, ma che di didattico non hanno nulla o quasi.

Per definire un sistema didattico, non è sufficiente avere una scheda CPU con della memoria RAM, un monitor, un display, sia esso a led o a 7 segmenti, ed infine una tastiera.

È necessaria una filosofia di sistema, che metta a disposizione dell'utente tutta una serie di caratteristiche, tali da permettere un apprendimento ordinato delle modalità operative dei microprocessori, primo passo per un utilizzo autonomo, anche in campo progettuale, delle conoscenze acquisite.

In prima approssimazione, gli elementi qualificanti di un sistema didattico sono:

- Configurazione base di semplice utilizzo
- Documentazione didattica, collegata al sistema, di alto livello
- Espandibilità verso configurazioni di più alto livello
- Modalità operative

Configurazione

La configurazione in genere proposta dai costruttori è costituita da:

- scheda microcomputer, con CPU, memoria RAM, socket per EPROM o PROM addizionali, ROM con monitor, interfacciamento parallelo/seriale
- tastiera ottale o esadecimale, sia incorporata nel sistema (sulla scheda) che a parte, collegata con un flat-cable alla scheda CPU (tipo hand-calculator)
- display, a led o a 7 segmenti, anch'esso o sulla scheda CPU o come parte della tastiera, la quale si configura quindi come una classica calcolatrice tascabile, naturalmente solo a livello estetico.

Con questa configurazione è possibile eseguire le seguenti operazioni:

1) Input dati da tastiera
2) Output risultati sul display

Questo, in una prima fase di studio, è più che sufficiente. Certo, così facendo, si sperimenta solo l'aspetto di LSI programmabile del microprocessore, tralasciando del tutto la sperimentazione hardware.

Spesso una variante è data dalla presenza di una basetta su cui poter realizzare circuiti, utilizzando segnali provenienti dal microprocessore stesso.

Nel caso del Nanocomputer® SGS-ATES, viene offerta la configurazione NBZ80-S, con basetta per sperimentazioni hardware.

La configurazione vista, con l'eventuale espansione hardware, rappresenta il minimo che si può pretendere da un sistema didattico.

Il sistema diventa poi vincente per tutta una serie di prestazioni, che sono funzione della potenza del monitor residente, della quantità di RAM disponibile, degli interfacciamenti previsti (cassetta, stampante, televisore casalingo, ecc.), in ultima analisi della operatività della tastiera.

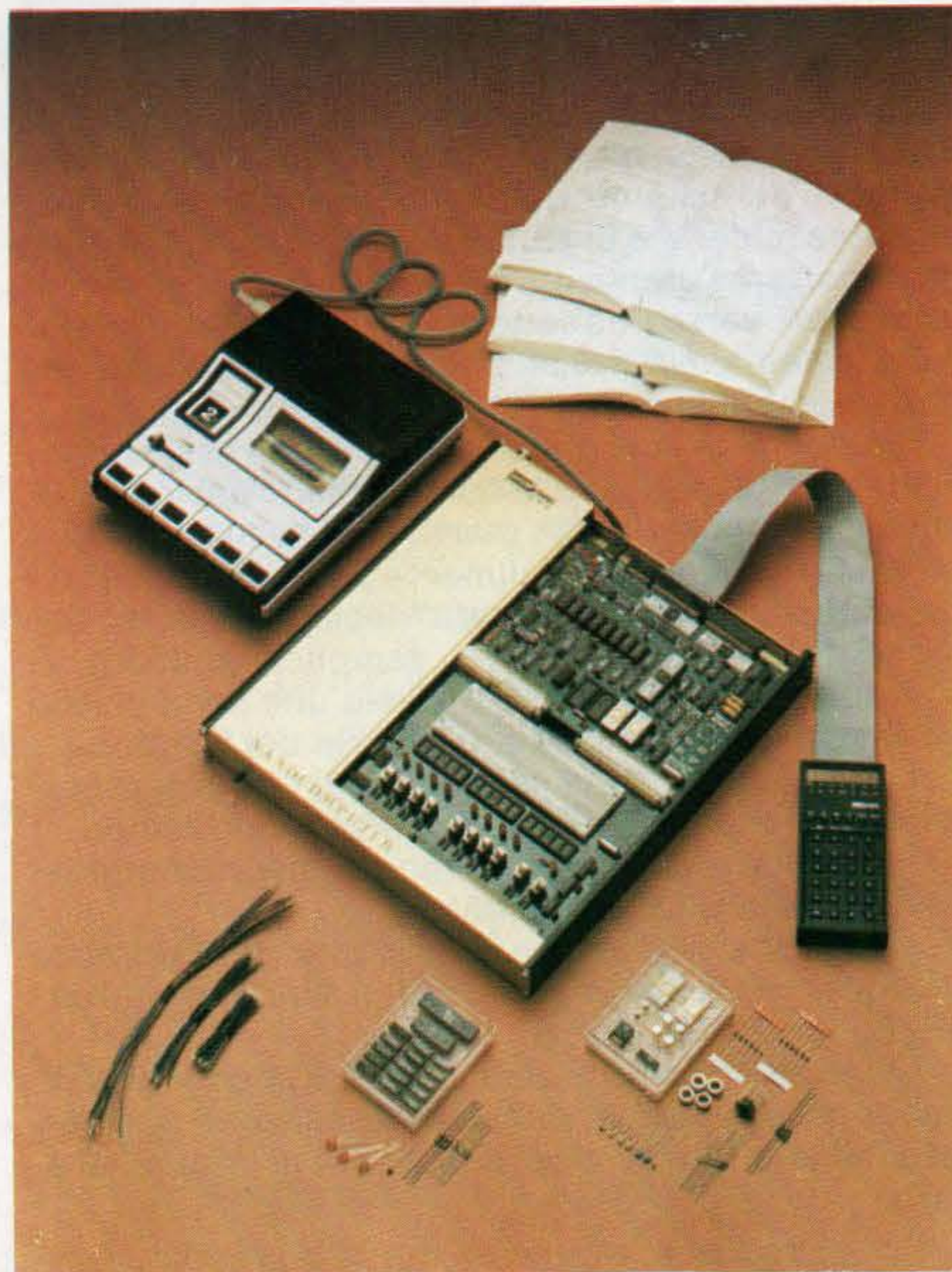



Figura 1 - Sistema completo Nanocomputer® realizzato dalla SGS-ATES. Si possono notare: tre volumi del Nanobook®, le due schede, base e per esperimenti (con bread-board), inserite nel contenitore, il registratore a cassette. Più in basso il kit di componenti per esperimenti ed il kit di espansione della scheda.

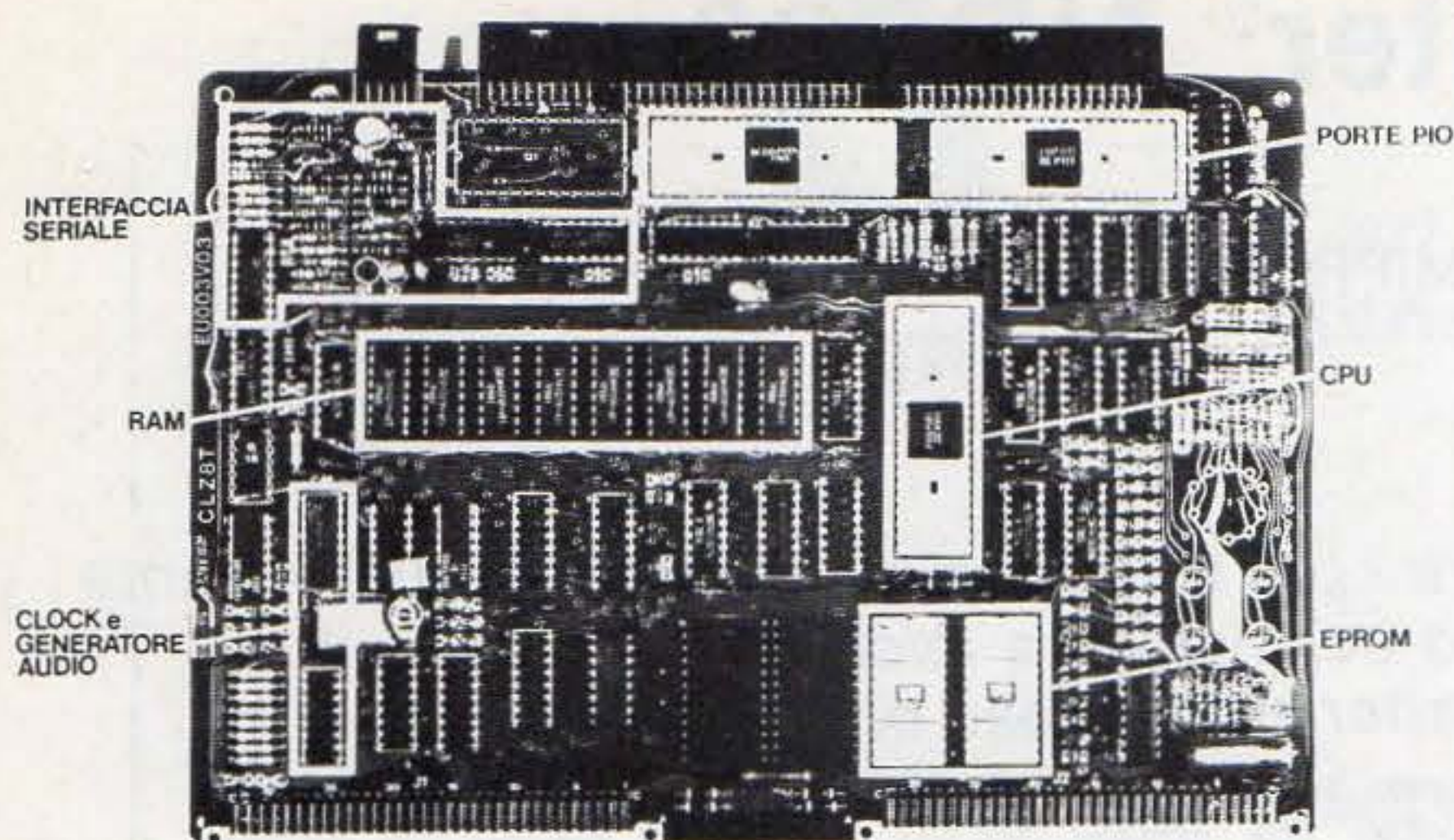



Figura 2 - Scheda del Nanocomputer® NBZ80 con evidenziate le funzioni più importanti.

Per l'aspetto hardware, è importante la presenza di un supporto sperimentale il più ampio possibile, come piccole schedine già pronte e attrezzature minime, forse banali per un progettista "di professione", ma di estremo ausilio per chi si accosta per la prima volta ai microprocessori.

Documentazione didattica

Indispensabile, se non addirittura pregiudiziale, è la presenza di ampia documentazione didattica.

Un sistema didattico è tale solo se possiede testi, esercizi, dispense, seriamente redatti, sia sull'uso del sistema, che, in generale, sulle modalità applicative di quel dato microprocessore, sempre nell'ambito del sistema didattico in oggetto.

Questa documentazione non va vista come una documentazione tecnica delle sole caratteristiche operative del sistema, quindi un manuale, ma piuttosto come materiale di studio di buon livello culturale, atto a portare l'utente oltre le basi generali dei microprocessori, verso una serie di sperimentazioni guidate sul sistema, per poi stimolarlo ad un uso indipendente personale di quanto ha appreso.

Quindi, al di là di quelle che possono essere considerazioni di merito sulla bontà o meno dei testi allegati al sistema didattico, questi testi devono esserci, altrimenti il sistema difficilmente potrà essere definito didattico, ma piuttosto sarà "declassato" ad evaluation kit, low-cost system, o semplicemente kit.

Attualmente sul mercato esiste una serie di testi didatticamente molto validi, la serie dei Bugbook.

Questa serie è stata creata sia per il microprocessore 8080 che per lo Z-80, in un secondo tempo.

I Bugbook 8080 supportano il microcomputer MMD-1 della E-L Instrument.

I Bugbook Z-80 invece sono stati scritti dagli stessi autori della serie 8080 per il Nanocomputer® NBZ80 della SGS-ATES.

Questi testi sono, al momento, tra i pochi testi didattici nel vero senso della parola esistenti sul mercato italiano (intesi, cioè, come testi di lingua italiana).

Espandibilità

Dopo un suo training personale, dopo aver sfruttato, o fatto sfruttare ad eventuali allievi tutte le possibilità del sistema didattico, l'utente non deve essere messo nella condizione di dover mettere da parte il suo sistema per acquistare qualcosa di diverso, posto che sia sua intenzione salire in complessità e sofisticazione di studio, di applicazioni, al limite se vuole tentare un primo progetto.

Il sistema didattico deve essere in grado di espandersi verso l'alto, "sparire" in una configurazione più ampia, di cui però sia sempre il nucleo centrale.

Allora, se il sistema è una scheda CPU con tastiera e display, è auspicabile poter eliminare la tastiera, inserire la scheda in un rack, collegare una tastiera ASCII, collegare un video, espandere la memoria, inserire un sistema operativo, interfacciare una stampante, un floppy.

Chiaramente questo può anche non essere, nel senso che ci si può accontentare di mantenere il sistema a livelli strettamente didattici: però deve poter essere possibile espandere la sofisticazione operativa del sistema in modo consistente.

Modalità operative

Le modalità operative sono ovviamente funzione della configurazione, sia base che proiettata in una futura espansione. Quindi, data per scontata la configurazione in genere proposta dai costruttori, precedentemente indicata, diamo una elencazione qualitativa dell'operatività auspicabile per un sistema didattico.

Nel dettaglio: deve esserci un single-step e l'utente deve abituarsi ad usare bene il single step; deve esserci la possibilità di inserire dei breakpoint e nella letteratura didattica deve esserci tutta una serie di esercizi atti a dimostrare, da un punto di vista di esigenze di lavoro industriale, come trarre il massimo profitto da un debug, fatto però con il breakpoint.

Deve poter essere possibile vedere tutti i registri del microprocessore in studio, senza ricorrere a tecniche software più o meno accettabili per porre questi registri in RAM e poi leggerli. Deve poter essere possibile interfacciarsi con una periferica più o meno evoluta, cassetta, TTY, con relativo software di gestione già presente nel firmware del monitor.

Ma il concetto base è il seguente: il sistema didattico deve avere un "sistema operativo" (monitor) che operi in modo da abituare l'utente alle reali procedure di sviluppo software, con l'obbiettivo di familiarizzarlo con gli strumenti base che troverà sempre e comunque in qualsiasi sistema di sviluppo per microprocessori.

Descrizione tecnica del Nanocomputer® NBZ80

Il Nanocomputer® NBZ80 (Fig. 1) si basa sulla CPU Z-80, microprocessore prodotto dalla SGS-ATES.

Sulla scheda (Fig. 2) è presente una memoria RAM di 4Kx8, realizzata con 8 chip M4027P, ed una memoria EPROM di 2Kx8 (2 EPROM M2708) in cui risiede il monitor NC-Z.

Il colloquio I/O è realizzato da due chip PIO, di cui uno gestisce l'interfaccia con la tastiera e con una cassetta audio, mentre l'altro è disponibile per l'utente, per la definizione di interfacciamenti personalizzati.

Il monitor NC-Z da 2K contiene le routine di gestione tastiera, come quelle di visualizzazione dati sul display a 7 segmenti ad 8 digit incorporato nella tastiera tipo hand-held (di gestione I/O verso terminali, tra cui appunto la cassetta magnetica).

general processor

INFORMAZIONE INTERNA PRELIMINARE
================================

Data: 1/6/79

Oggetto: entrata in produzione del nuovo sistema "T"

Dalla pluriennale esperienza della General Processor nel settore dei personal computers nasce oggi un nuovo sistema destinato ad impieghi di carattere hobbistico o professionale ove la compattezza, le prestazioni, il basso costo siano i fattori principali della scelta.
Il nuovo "MODELLO T", progettato e prodotto in Italia, ricalca alcuni elementi tipici della produzione GP, come l'unita' centrale Z-80, software compatibile con il microprocessore 8080/8085, e le memorie dinamiche di estese dimensioni, ma e' stato completamente ridisegnato rispetto ai precedenti modelli, sia nella parte circuitale che nella stessa filosofia di sistema. Il contenitore racchiude adesso i principali elementi funzionali dell'insieme: l'unita' centrale, la memoria, con possibilita' di espansione fino a 48K RAM e 12K epROM, l'interfaccia per un monitor video con set esteso di caratteri, un minifloppy disk drive IBM compatibile, una stampante termica a 20 colonne completamente alfanumerica, l'alimentatore dell'insieme ed una tastiera alfanumerica a 76 tasti di tipo professionale con keypad numerico separato. La forma del contenitore e' poi tale da consentire l'appoggio superiore del monitor televisivo in modo da realizzare un insieme compatto e di facile uso. Cinque "slots" a disposizione per l'utente permettono poi l'inserzione di schede accessorie per il collegamento di un comune registratore audio, di una stampante veloce a carrello largo, di altre unita' a disco o di periferiche utente seriali o parallele. Prezzi a partire da 1.549.000. Ulteriori informazioni ed il listino prezzi completo possono essere richieste alla GP o ai suoi rivenditori:
CM ELETTRONICA - Via Pegaso 48 - SOTTOMARINA DI CHIOGGIA (VE)
TECEM - Via IV Novembre, 48 AREZZO
Studio Commerciale dr. Tripodi - Via Negrelli - S.Giovanni in Fiore (CS)
I prodotti della General Processor godono di una ampissima diffusione ed essendo progettati e costruiti in Italia fruiscono della migliore assistenza immaginabile. I prototipi del "MODELLO T" saranno esposti a Milano alla prima rassegna del personal computer (Fiera, 6-9 giugno).

Via Montebello 3R - Firenze Tel. (055) 21.91.43

Il monitor del Nanocomputer® NBZ80

In tutti i sistemi, uno degli elementi più qualificanti è il software residente, responsabile dell'operatività del sistema, delle sue prestazioni,della sua comodità di utilizzo. Il firmware dell'NBZ80 è indicato con NC-Z, ed è contenuto in due EPROM 1Kx8 2708.

In termini di espandibilità, i due socket dell'NC-Z ed i due liberi sull''NBZ80 possono essere occupati da 4 EPROM 2716, 2Kx8, costituenti il monitor 8K FR-Z, con Editor, Edit, Assembler. Il monitor base NC-Z è molto sofisticato soprattutto in termini di operatività da tastiera (Fig. 3).

I comandi che l'utente può dare all'NBZ80 sono suddivisibili in 5 gruppi:

1) gruppo DATA
Con i tasti numerici da 0 a F si possono introdurre i 16 caratteri esadecimali
2) gruppo DATA DISPLAY
Questo gruppo è costituito da 4 tasti: "freccia a destra" e "freccia a sinistra", per lo spostamento dell'indicazione luminosa in corrispondenza dell'entità il cui contenuto vuole essere visualizzato; 2ND, per il caricamento del byte HIGH in una delle coppie di registri BC, DE, HL, AF; ARS, per la visualizzazione dei contenuti dell'Alternate Register Set.
3) gruppo PROGRAM CONTROL
Di questo gruppo fanno parte 5 tasti: BRK, per la definizione dei breakpoint, cioè dei punti di arresto nel corpo del programma utente (sono possibili fino a 8 punti di arresto); SS, o Single-Step, per l'avanzamento passo passo del programma; GO, per lo start del programma; BREAK, per realizzare l'arresto di un programma e restituire il controllo al monitor NC-Z; RESET, per la reinizializzazione globale dell'NBZ80.
4) gruppo PROGRAM ENTRY
Questo gruppo prevede 3 tasti: LA, o Load Address, per il caricamento di un indirizzo di memoria precedentemente digitato con i tasti esadecimali; ST, o Store, per la memorizzazione dei dati nella locazione di memoria voluta, o nei registri indicati; INC, o Increment, per incrementare di 1 l'indirizzo della zona di memoria sotto osservazione.
5) gruppo DUMP-LOAD
Questo gruppo prevede due soli tasti, il tasto di DUMP (DP) per scaricare su cassetta o su un terminale seriale (TTY) il contenuto della RAM, e LOAD (LD) per caricare da cassetta o TTY verso la memoria RAM dell'NBZ80.

Espansione del Nanocomputer® NBZ80

Il Nanocomputer® è espandibile verticalmente da una configurazione base ad una più complessa, come già detto.

Questa espansione si realizza su diversi fronti: schede, periferiche, software, accessori vari.

Prima di tutto, l'NBZ80 può diventare un microcomputer CLZ80 - 4/2 con il Kit di conversione KNZ80, contenente anche il nuovo monitor MO-Z.

Successivamente, la scheda microcomputer può essere continuata da una serie di altre schede:

1) RAZ80, scheda di memoria da 16,32 o 48K RAM
2) PIZ80,scheda di I/O con I/O parallelo 8x8 bit, due I/O seriali, 4 canali counter-timer

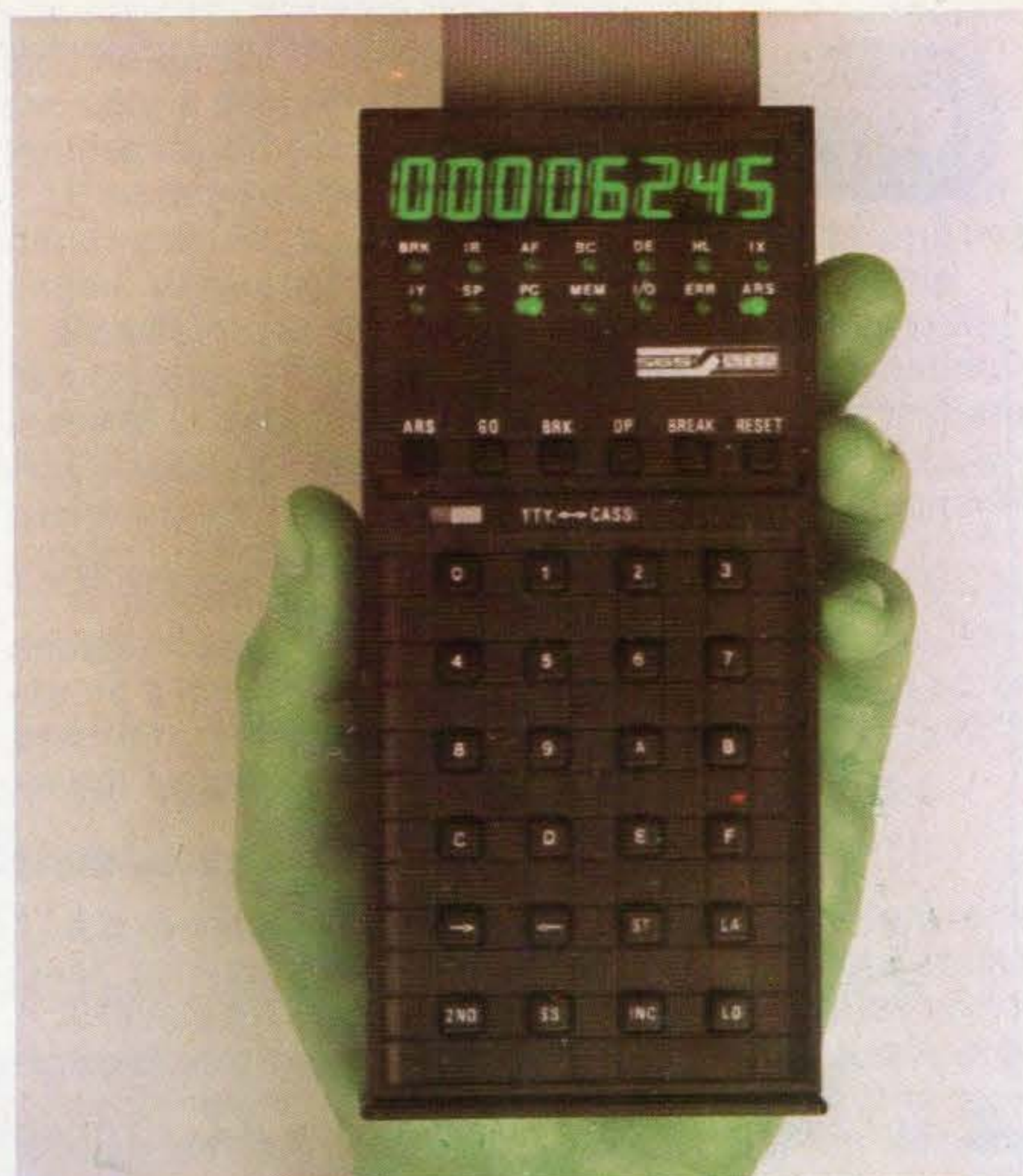



Figura 3 - Tastiera e display del Nanocomputer® NBZ80.

3) VDZ80, scheda interfaccia video
4) FLZ80, scheda interfaccia Floppy-disk
5) PPZ80, scheda PROM Programmer

A questo si aggiungano le periferiche, terminale VDU (VTZ80), video display (TVZ80), stampante seriale (SSZ80).

Il tutto inseribile negli slot appositamente forniti, CTZ80, card cage ad 8 slot, CPZ80, a 4 slot.

Questa linea di espansione quindi appare, almeno potenzialmente, valida.

La potenzialità accennata si basa sul fatto reale che ancora non è stata realizzata un'ampia sperimentazione da parte dell'utenza, per cui non si è ancora in grado di dare un giudizio definitivo sulla linea proposta dalla SGS-ATES.

Sempre nell'ambito dell'espandibilità, il software appare molto curato:

- MO-Z, Monitor/Debug da 2K
- FR-Z, Monitor/Editor, Assembler da 8K
- ASS-Z, Assembler da 4K, su cassetta
- EDI-Z, Edit da 2K, su cassetta
- EPR-Z, Prom programmer da 1K, su PROM o cassetta
- BAS-Z, Basic da 8K, su PROM.

Conclusione

Questa prima presentazione del Nanocomputer® NBZ80 lascia senz'altro intravedere ottimi sviluppi di questo sistema che, fin dal suo primo apparire sul mercato, ha suscitato un indiscusso interesse.

Il successo sarà funzione del supporto che il sistema riceverà da parte della SGS-ATES, oltre che, ovviamente, dall'accoglienza che gli sarà riservata dall'utenza.

Al fine di permettere una prima valutazione, a questo articolo farà seguito una piccola serie appunto finalizzata ad un utilizzo critico dell'NBZ80, per individuarne pregi e difetti. ■

E' DURO PER ALCUNI DOVER AMMETTERE DI AVER PERSO IL "QUASI MONOPOLIO". I FATTI SONO QUELLI CHE CONTANO:

LA REALTÁ DEI MICRO COMINCIA DALLA "Z"

COMPONENTI MICRO

PIASTRE

CALCOLATORI E SISTEMI

SOFTWARE E SUPPORTO

DUE FISIONOMIE DIVERSE MA UNITE SOTTO MOLTI ASPETTI

Zilog

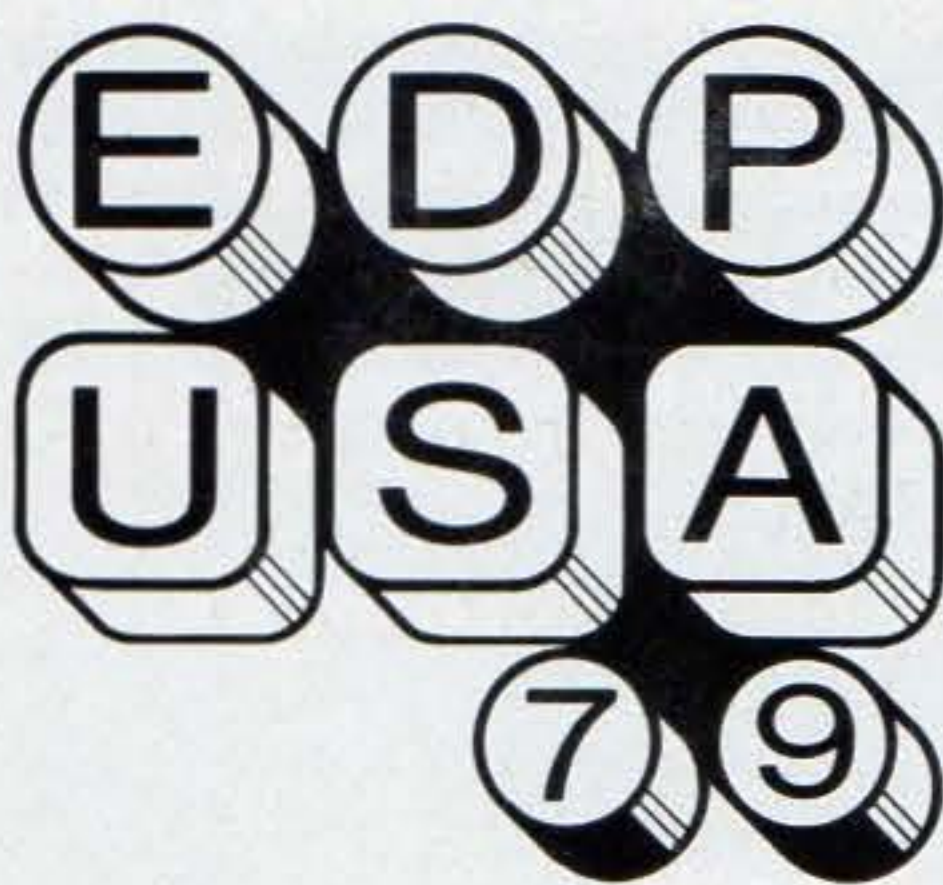
- E' l'unica casa completamente dedicata ai micro.
- Progetta tenendo presente i problemi globali ed in funzione di questi ottimizza i particolari perciò:
- vi offre di meglio come componenti, come piastre, come sistemi e come software.
- Ha alle spalle il più forte gruppo finanziario del mondo;
- ciò le consente di avere i migliori specialisti mondiali di hardware, di software e di sistemistica e da
- la sicurezza di disporre dei finanziamenti necessari a conservare la supremazia nel proprio campo (per questo Vi sono ditte anche in italia, che già nel 1979 hanno impostato la produzione di centinaia di migliaia di macchine con lo Z80).

Zelco

- E' esclusivamente dedicata ai micro e dà un supporto che va dalla fornitura di componenti, agli accessori (terminali, stampanti etc.) per sistemi
- E' composta da specialisti e non da venditori per offrire un servizio più rapido e non una assistenza che si ferma di fronte ai problemi nuovi;
- Usa essa stessa intensamente i calcolatori che vende, li conosce a fondo e produce software sofisticato per se e per gli altri.
- Ha una organizzazione agile che le permette di non ricaricare spese eccessive sui prodotti che vende e di rispondere in "real time". (Provare per credere).
- Ha una consegna pronta; ciò significa che non dovete aspettare due mesi per un SIO o una CPU e neppure per un calcolatore.
- Vi assiste direttamente per riparazioni o consulenza qualificata.

La nostra pubblicità maggiore è fatta da tre elementi:

1 - I nostri clienti **2 - La nostra qualità** **3 - I nostri prezzi**

Fate i confronti: per noi basta telefonare al 02/803336/804247 (nelle zone di Roma al 06/8102836)

General Processor: modello T

di **Gianni Becattini** - *Direttore Generale della General Processor/Processor Sud*

Premessa

Il nome General Processor non è certo ignoto a chi si interessa di microelaboratori. La GP è infatti la prima azienda italiana che si sia dedicata alla produzione di elaboratori personali avendo iniziato la sua attività esattamente tre anni or sono.
In questo periodo ha prodotto diverse centinaia di sistemi denominati Child 8/BS e Child Z basati rispettivamente sui microprocessori F8 e Z-80.
Recentemente è stato annunciato dalla GP un nuovo sistema, il "MODELLO T".
Siamo lieti di presentarne in anteprima assoluta ai lettori di BIT le caratteristiche fondamentali.
Il nuovo "MODELLO T" è un microelaboratore integrato destinato alle applicazioni hobbystiche e professionali dove una elevata potenza di elaborazione ed una eccellente versatilità non debbano essere disgiunte dalla compattezza delle dimensioni e dal costo limitato.
Dentro il contenitore del 'T' sono racchiusi quasi tutti i principali elementi funzionali: l'unità centrale, la memoria, l'interfaccia video, l'interfaccia per audiocassette, un minifloppy-disk IBM compatibile, una stampante termica, l'alimentatore ed una tastiera professionale con "pad" numerico separato.
Al di sopra del contenitore stesso può essere appoggiato un monitor televisivo in modo da realizzare un insieme compatto e di facile uso. Interfacce seriali e parallele consentono il collegamento con periferiche esterne (Fig.1)

La filosofia del progetto

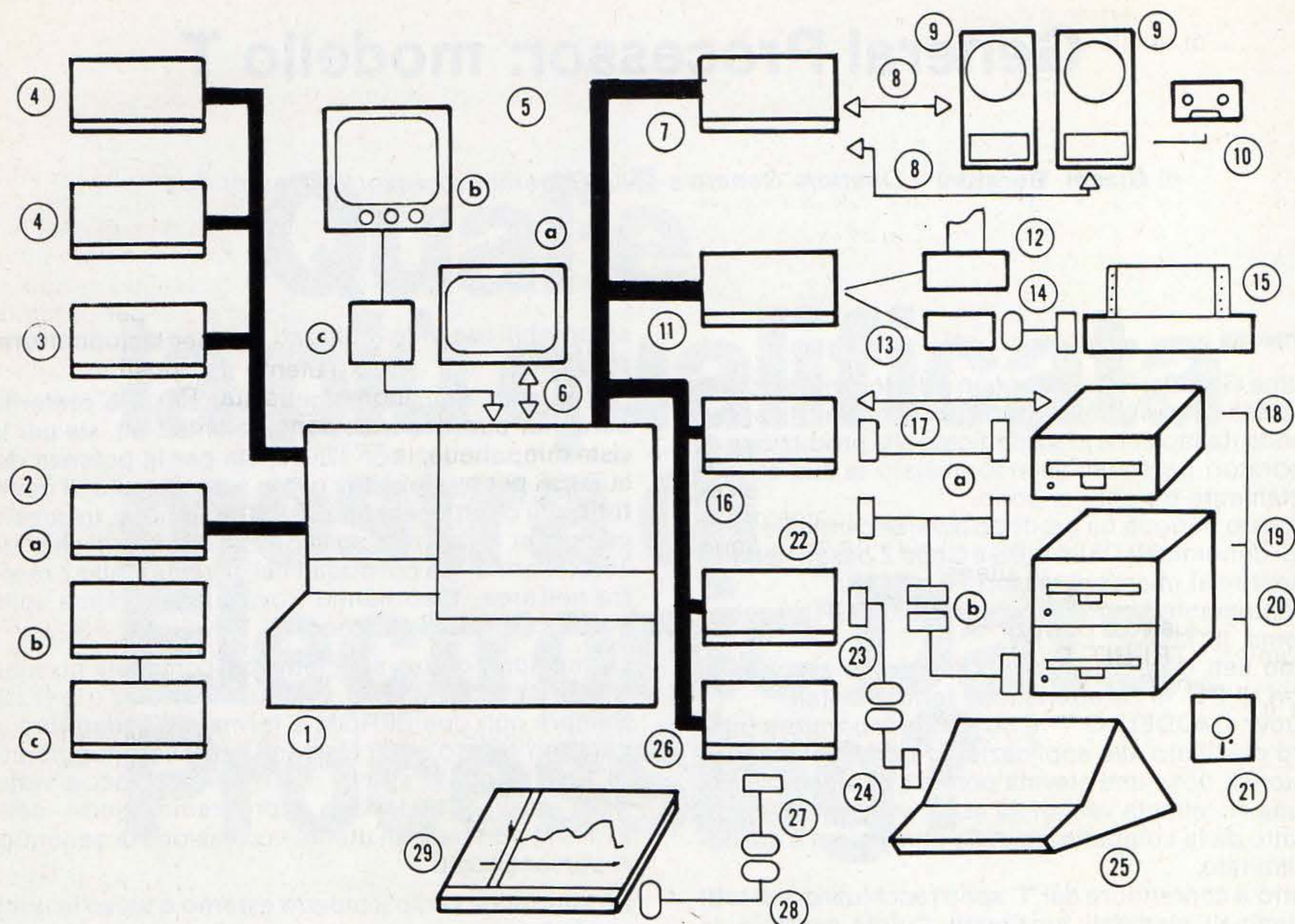
Uno dei principali obbiettivi del progetto T è stato quello di concentrare in una porzione di spazio limitata la maggiore potenza di elaborazione senza ricorrere a tecniche costruttive particolari come i circuiti stampati multi strato. Lo scopo è stato ottenuto impiegando un concetto molto simile a quello dei televisori a colori: una grande scheda stampata di tipo tradizionale in cui si inseriscono varie minischede tramite connettori verticali (Fig.2).
In virtù di questa disposizione il collaudo, la manutenzione e l'espansione del sistema risultano di considerevole facilità, anche perchè previa la semplice asportazione del cofano superiore si guadagna l'accesso ad ogni elemento del circuito. L'aggiunta di elementi accessori risulta così talmente semplice da poter essere tranquillamente effettuata anche da parte dell'utente meno esperto.
La manutenzione è avvantaggiata e viene fatta o a livello di integrato (sono tutti montati su zoccoli) o di minischeda. I collegamenti in filo sono rapidamente smontabili essendo realizzati a mezzo di connettori a pressione.

Per quanto riguarda l'unità centrale si è preferito continuare a dare la preferenza allo Z-80, sia per la sua compatibilità con l'8080, sia per la potenza del suo set di istruzioni, sia per la sua capacità di interfacciarsi direttamente alle RAM dinamiche. In questo campo si è fatto esclusivo uso dei nuovi chip da 16Kx1, ormai già collaudati nel sistema Child Z mentre nell'area video hanno trovato per la prima volta impiego le 1Kx4 statiche.

La memoria di una sola lettura è composta normalmente da EPROM 1Kx8, sostituibili con le 2 o le 4Kx8. Sempre con due EPROM è formato il generatore di caratteri esteso a 128 elementi e matrice maggiorata di 13x8 punti, lo stesso usato nell'interfaccia video della serie Child. Una riprogrammazione delle EPROM permette all'utente la creazione di generatori personalizzati.

Le interfacce verso il mondo esterno o verso le unità periferiche di sistema sono realizzate con circuiti MOS-LSI della serie Z-80 od 8080 con l'eccezione del floppy disk controller, il 1771 della Western Digital, che lavora in formato IBM.

Il contatto uomo/macchina

Il contatto uomo/macchina avviene nel modello T, come nella maggior parte degli elaboratori di questo tipo, attraverso una tastiera alfanumerica ed uno schermo televisivo; per questo motivo queste parti sono state molto curate.
La tastiera conta ben 73 tasti ed è di classe professionale, completamente priva di contatti elettrici, con vita garantita di oltre 100 milioni di operazioni.
Le scritte, per evitare che l'usura possa danneggiarle, sono incorporate a spessore nella plastica del tasto. Alcuni tasti di controllo permettono il funzionamento in vari modi funzionali mentre la sezione algebrica assicura una rapida e comoda impostazione di quantità numeriche analogamente a quanto avviene con le calcolatrici.
La sezione video genera sullo schermo una nitida e stabile pagina di 16 righe di 64 caratteri ciascuna. I caratteri possibili sono 128 tra cui le lettere minuscole, maiuscole, i segni grafici e matematici ecc. Un puntatore spostabile tramite appositi tasti indica la posizione operativa in corso. Per ogni elemento è possibile stabilire il contrasto positivo o negativo, ossia la scrittura in bianco su nero o viceversa; grazie a ciò risulta semplificata la creazione di maschere e di diagrammi. Una diversa programmazione del generatore di caratteri, che implica alcune rinunce sulle

1 - Microcomputer base
2 - Scheda di memoria in sola lettura (EPROM)
- a - 4K (4 × 2708) - in dotazione standard
- b - 8K (4 × 2716)
- c - 12K (3 × 2732)
3 - Scheda di memoria in lettura/scrittura (RAM) - in dotazione standard
4 - Schede di estensione di memoria in lettura/scrittura da 16K ciascuna
5 - Video display
- a - monitor standard
- b - televisore utente interfacciato direttamente o tramite
- c - modulatore video per ingresso da presa antenna TV

6 - Cavo di connessione video display
7 - Interfaccia per due audio registratori
8 - Cavi di connessione registratori
9 - Audio registratori
10 - Cassetta per programmi o dati
11 - Scheda di interfaccia parallela TPIO
12 - Stampante termica alfanumerica a 20 colonne
13 - Interfaccia per stampante serie 700
14 - Cavo di connessione per stampante serie 700
15 - Stampante alfanumerica a pagina serie 700
16 - Floppy disk controller IBM compatibile
17 - Set di cavi interni per la connessione del minifloppy disk drive
- a - per la connessione del solo floppy interno
- b - per la connessione dei floppy interno ed esterni

18 - Minifloppy disk drive interno
19 - Primo minifloppy disk drive esterno con cabinet ed alimentatore
20 - Secondo minifloppy disk drive esterno
21 - Minidisco per programmi e dati
22 - Scheda di interfaccia parallela TPIO per la connessione di una periferica non standard (questo spazio potrebbe essere occupato invece da una scheda TSER)
23 - Cavo DIP/DF di connessione della scheda TPIO al pannello posteriore del microcomputer
24 - Cavo DM/DIP di connessione del pannello posteriore alla periferica esterna
25 - Periferica esterna non standard con interfacciamento parallelo
26 - Scheda di interfaccia seriale TSER per la connessione di una periferica esterna che usi lo standard RS-232-C (opp. 20mA loop di corrente). (1)
27 - Cavo 2DIP/DIM di connessione dalla scheda TSER al pannello poteriore del microcomputer
28 - Cavo DM/DM di connessione dal pannello posteriore alla periferica esterna RS-232-C (opp. 20mA loop di corrente)
29 - Periferica esterna interfacciata serialmente sullo standard RS-232-C (opp. 20mA loop di corrente). Nella figura in plotter per uscita grafica.

(1) - Questo spazio potrebbe essere occupato da una scheda TPIO

Figura 1 - Il sistema T nei suoi elementi.

maiuscole e sui simboli matematici, conferisce alla macchina la capacità di tracciare grafici con risoluzione di 48 × 128 punti.
Un altro importante punto di contatto tra l'uomo e la macchina è la stampante. Si tratta di un accessorio essenziale perchè fissa su carta i risultati del lavoro compiuto o la lista di un programma.
Gli utenti del modello T dispongono di due alternati-

ve: l'installazione di una stampantina a striscia di carta internamente al contenitore od il collegamento di una stampante seriale a foglio di grandi dimensioni come una di quelle della serie 700.

Le memorie

La memoria centrale del modello T può essere espansa fino a 48K di RAM (lettura/scrittura) e 12K di EPROM (sola lettura), oltre al K di pertinenza alla sezione video. Solitamente il sistema T è basato più su una notevole quantità di RAM che non di EPROM. I vantaggi che ne derivano consistono essenzialmente in una molto maggiore versatilità che implica la possibilità di adoperare diversi linguaggi o sistemi operativi in funzione dello scopo prefisso.
Come memoria di massa l'utente può impiegare le audiocassette od i floppy disk.
Le cassette godono del principale vantaggio dell'economia: l'interfaccia è veramente poco costosa e come registratore può essere usata una qualunque unità in commercio di caratteristiche appena accettabili. La registrazione viene effettuata secondo un metodo esclusivo a correzione automatica di errore denominato TRI-BIT. Data la ridondanza di informazioni, il flusso effettivo dei dati avviene a circa 800 bit/sec, il che implica che per caricare 16K sono richiesti solo due minuti e mezzo circa. Il motore di trascinamento nastro di due mangiacassette può essere controllato dal programma.
I floppy disk rappresentano un mezzo dalle possibilità infinitamente superiori nei confronti delle cassette. La loro caratteristica fondamentale è quella di essere dei dispositivi ad accesso casuale e non sequenziale; vale a dire che da essi le informazioni possono essere scritte o lette in qualsiasi ordine mentre nel nastro è necessario svolgere tutta la parte di bobina che precede il punto interessato prima di poter scrivere o leggere. Il disco opera poi scambi a velocità enormemente superiore al nastro con risparmi di tempo veramente sostanziali specie nella gestione di archivi, magazzini ecc.
Un minifloppy disk può essere installato dentro il contenitore del "T" ed altri due "drivers" possono essere collegati esternamente. Il minifloppy ha una capacità di circa 100K per dischetto e lavora su formato IBM.
Per applicazioni di maggiore impegno la GP ha previsto la connessione di unità di floppy disk standard, sempre IBM compatibili, da 256K ciascuna.

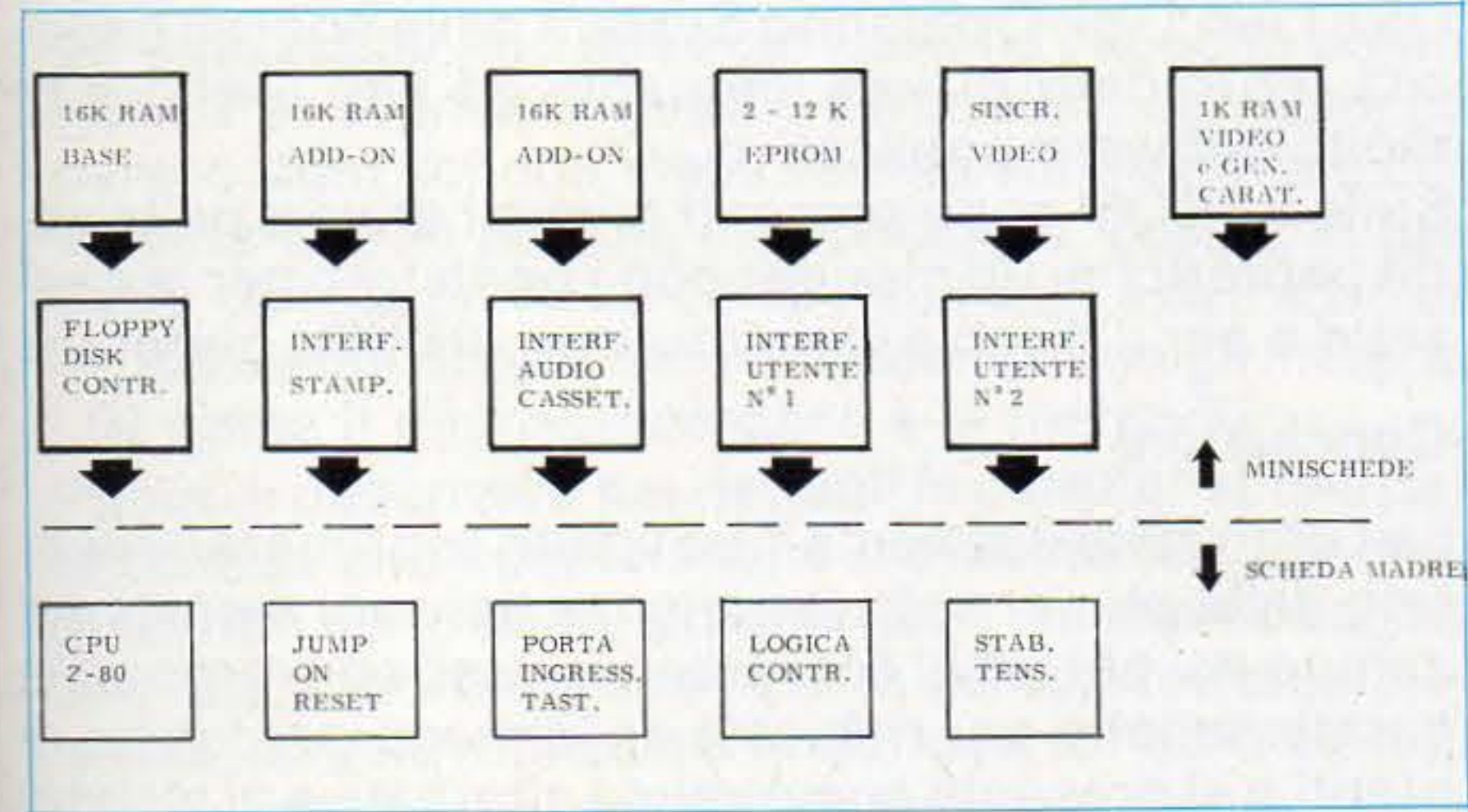



Figura 2 - Distribuzione dei circuiti all'interno del modello T.

Il disco costituisce forse uno dei maggiori punti di forza del modello T distaccandolo nettamente da altri modelli similari non provvisti di tale prerogativa.

Il sistema di ingresso/uscita del modello T

Un altro punto molto interessante del modello T è la sua potente e versatile capacità di essere interconnesso con le periferiche più diverse. Nel contenitore possono essere alloggiate cinque schede di interfaccia in appositi connettori già predisposti nella versione base. In un sistema al suo massimo grado di espansione, dotato cioé di interfaccia per cassette, floppy disk controller, interfaccia per stampante, restano ancora liberi per l'utente due posti per altrettante schede di ingresso/uscita. Per l'interfacciamento di tipo generale sono state al momento previste due schede, la TPIO per collegamenti paralleli e la TSER per quelli seriali. Ogni scheda dispone di vari tipi di cavi per facilitare l'utente nel collegamento. Altre schede possono essere realizzate dall'utilizzatore per coprire anche la più particolare fascia di applicazione.
La scheda TPIO provvede due porte parallele di 8 bit ciascuna più 4 linee destinate a sincronizzare il flusso dei dati. Ogni bit può essere programmato per fungere come ingresso o come uscita ed è ammesso anche il modo bidirezionale. La scheda TPIO inoltre, in virtù di un circuito integrato a larga scala in essa impiega-

- Numeri di linea da 1 a 63200
- Aritmetica intera e floating point in semplice e doppia precisione
- Fino a 17 cifre significative
- Variabili simboliche alfanumeriche
- Stringhe con dimensionamento automatico e matrici anche di stringhe
- Operazioni logiche
- Possibilità completa di editing
- Diagnostica estesa degli errori
- Operatori per il controllo della stampante

COMANDI

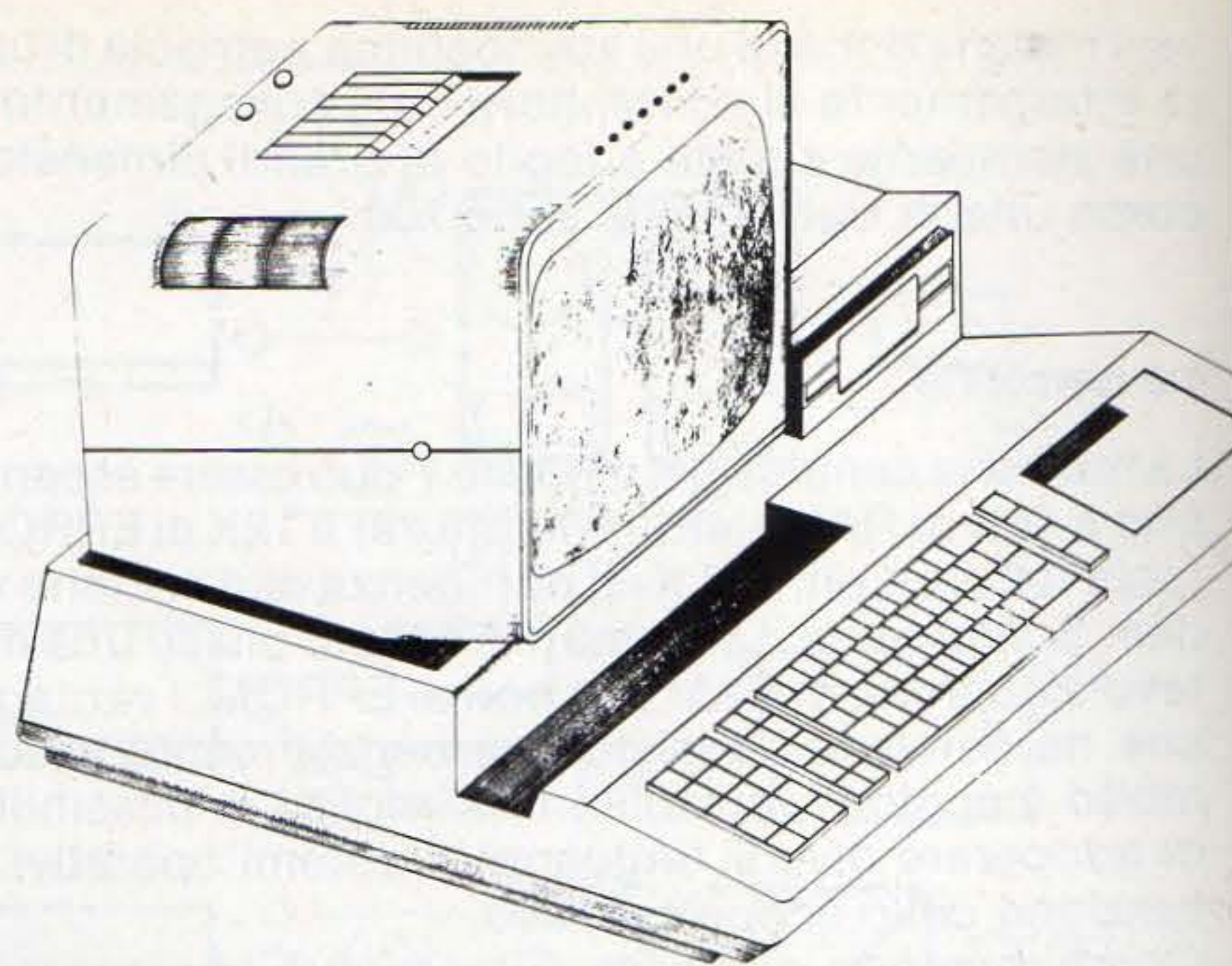
CLEAR	CLOAD	CONT	CSAVE
DELETE	EDIT	LIST	LLIST
NEW	RUN		

FRASI

DATA	DEF	END	DIM
FOR	GOTO	GOSUB	IF . . THEN
IF . . THEN . . ELSE	INPUT	LET	LPRINT
LPRINT USING	MID$	NEXT	ON ERROR GOTO
ON . . GOTO	ON . . GOSUB	OUT	POKE
PRINT	PRINT USING	READ	REM
RESTORE	RESUME	RETURN	STOP
SWAP	TROFF	TRON	WAIT

FUNZIONI

ABS	ASC	ATN	CINT
CSNG	CDBL	CHR$	COS
ERL	ERR	EXP	FIX
FRE	HEX$	INP	INSTR
INT	LEFT$	LEN	LOG
LPOS	MID$	OCT$	RND
POS	RIGHT$	SGN	SIN
SPACE$	SPC	SQR	STR$
STRING$	TAB	TAN	VAL
PEEK			

Figura 3 - Principali caratteristiche dell'Extended BASIC.

to, può sollevare l'unità centrale da alcuni lavori di monitoraggio delle linee di ingresso. E' possibile per esempio generare una interruzione quando due o più linee o qualcuna di esse raggiunge un determinato stato. E' facile intuire come questa capacità risulti estremamente preziosa ai fini della resa complessiva del sistema.
La scheda TSER invece è stata progettata per interfacciarsi con le numerose periferiche del commercio che impiegano il metodo standard RS-232C o quello molto simile a loop di corrente di 20mA. Tra di esse citiamo la quasi totalità di terminali e stampanti, i plotter e la strumentazione scientifica. La scheda TSER può operare in modo asincrono o sincrono e può essere usata anche per collegarsi direttamente o tramite modem ad un altro elaboratore in qualità di terminale a basso costo e di notevolissime prestazioni.

Il software

Il modello T viene attualmente dotato di due interpreti BASIC e di un debugger esadecimale, i primi su RAM ed il terzo su EPROM. Il primo dei due interpreti BASIC occupa solo 2K ed è destinato a risolvere i problemi che necessitino di una notevole velocità di elaborazione, pur essendo ovviamente alquanto limitato. Il secondo è invece un interprete esteso con molte funzioni che si trovano di consueto solo in sistemi di costo molto superiore. L'Extended BASIC (Fig. 3) occupa 14.5K RAM e comprende anche un editor nel suo interno.
Sul disco può essere ottenuta una versione adattata di CP/M (Marchio depositato della Digital Research, USA), un Disk Operating System dalle enormi capacità, corredato di vari linguaggi ad alto livello e di molto software applicativo.
Sulle EPROM sono presenti anche i drivers delle varie periferiche, un monitor con i bootstrap per le cassette e per il disco e varie routines di utilità generale.

Conclusioni

La GP tramite il sistema T ha voluto esprimere il risultato della pluriennale esperienza dei suoi tecnici nel campo dei personal computer. In esso convergono le caratteristiche più richieste dalla maggiore fascia di utenti e si presenta come valida alternativa ai sistemi di importazione od a quelli di costo assai più elevato. Le prime consegne sono previste per settembre. ■

AMICO 2000: un microcomputer anche in Kit

di **G. Ghiringhelli**

Quando si parla di microcomputer, siano essi su scheda che assemblati sotto forma di home-computer, si associa automaticamente a queste macchine il nome di un costruttore californiano o giù di lì. A questo ci siamo abituati dato che là, negli U.S.A., tutto comincia e tutto si rinnova in questo affascinante settore dell'elettronica.

Una volta tanto le cose cambiano: il tema è ancora il microelaboratore e l'home-computer, ma il nome del sistema AMICO 2000 è tutto italiano, come pure la società costruttrice A.S.E.L. s.r.l. di Milano.

Il sistema a microelaboratore AMICO 2000 è stato concepito secondo una filosofia "a crescita totale" in modo da interessare un amplissimo spettro di utilizzatori: a cominciare dall'hobbista principalmente interessato all'aspetto hardware del microprocessore, per finire con il sistemista che necessita di blocchi funzionali già pronti da utilizzare o di un sistema completo cui aggiungere solo la parte software.

Per rispondere a queste ambiziose finalità, il sistema è stato strutturato in modo da essere completamente modulare.

Ogni blocco infatti è indipendente dall'altro nel senso che ogni configurazione intermedia a partire dalla minima fino alla massima può vivere da sola, essendo utilizzabile fino alla sua massima capacità potenziale. Per scendere in altri particolari, ogni blocco, ovvero ogni scheda viene fornita sia montata e collaudata che in scatola di montaggio, cosa che permette da una parte un certo risparmio e dall'altro dà la possibilità di prendere confidenza con gli integrati LSI come il microprocessore e le memorie.

Prima di descrivere nei dettagli le caratteristiche delle schede che costituiscono il sistema (alcune delle quali non sono ancora disponibili) vorremmo segnalare una importante scelta fatta dal costruttore e cioè quella di aver utilizzato le schede formato Europa che, oltre ad essere molto compatte e maneggevoli, (mm 160x100) offrono la possibilità di essere inserite in contenitori a rack standard per questo formato.

L'unica scheda che come dimensione si discosta da questo formato è quella base che contiene il microprocessore; questa scelta, come si vedrà, è giustificata dal fatto che la scheda è stata studiata per poter vivere anche da sola.

Partendo da questa scheda il sistema espanso alla sua massima capacità diventa un potente microelaboratore che utilizza come display un normale televisore domestico (o un adatto monitor TV)

Il cuore del sistema è un 6502

La scheda principale che contiene la CPU è denominata AMICO 2000A ed è basata sul microprocessore a 8 bit 6502, una CPU bene affermata sul mercato, soprattutto per le sue eccellenti capacità di indirizzamento, e fornita da diverse case. L'AMICO 2000A ha una configurazione circuitale piuttosto classica, ma una compattezza fuori dal comune per un microelaboratore completo: le dimensioni della scheda infatti sono di 300x100mm (v. fig. 1).

Questa scheda, come si è accennato, è autosuffi-

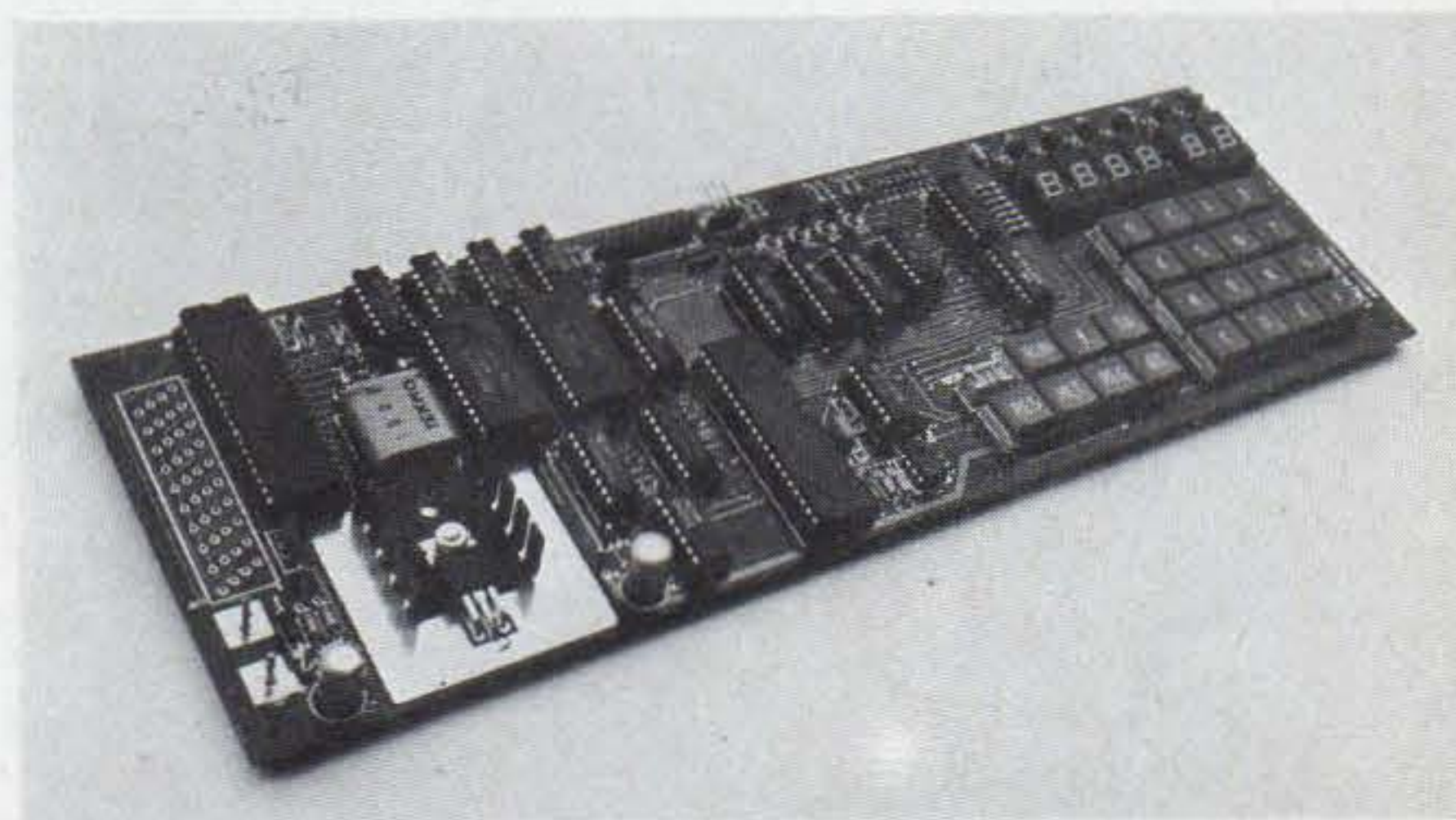



Figura 1 - La scheda base del sistema AMICO 2000 basata sul microprocessore 6502 è fornita sia montata e collaudata che in scatola di montaggio.

ciente. Essa permette tutte le operazioni di input di dati, attraverso una tastiera esadecimale ed alcuni tasti funzionali, mentre per il controllo delle operazioni esiste un display a sette segmenti LED con quattro cifre per visualizzare gli indirizzi di memoria e due per i dati.
I tasti funzionali dell'AMICO 2000A sono l'AD che permette di selezionare l'indirizzo della locazione di memoria che si intende esaminare o modificare; il tasto DA che permette di modificare il contenuto di una locazione di memoria; il tasto di incremento che consente di esaminare la locazione successiva a quella sulla quale siamo posizionati; il tasto REG che consente l'indirizzamento immediato di una locazione di memoria precedentemente selezionata; il tasto RUN che fa partire l'esecuzione di un programma; il tasto RES che ha la funzione di inizializzare il sistema dell'accensione, posiziona il program counter alla locazione di memoria 0000 e permette di arrestare l'esecuzione di un programma utente in qualsiasi momento, passando il controllo del sistema al monitor. L'ultimo azionamento presente sull'AMICO 2000A è un interruttore che consente il funzionamento in passo singolo (single step). La scheda è completa di un circuito per l'interfacciamento diretto con un registratore a cassetta; sono associati inoltre 4 LED la cui accensione identifica le fasi di registrazione e lettura della cassetta magnetica. A disposizione dell'utente infine c'è una porta di I/O a 8 bit per comunicare con l'esterno.
La memoria residente sull'AMICO 2000A è costituita da 2kbyte di RAM statica (4 RAM tipo 2114) e 1K byte di firmware su 2 PROM (tipo 93448) sul quale risiedono il monitor e il programma di gestione della cassetta.
L'indirizzamento della memoria e dell'I/O della scheda è realizzato su due PROM (tipo 74S287) che permettono una rapida ridefinizione del sistema. Sull'AMICO 2000A inoltre tutti i segnali di controllo e di comando sono presenti su un connettore a 40 poli per cavo piatto che viene utilizzato per l'espansione dell'intero sistema.
Per alimentare questa scheda è sufficiente una tensione raddrizzata e filtrata compresa fra gli 8 e i 12 Volt non stabilizzati e una corrente fino a circa 900mA: L'AMICO 2000A infatti incorpora un regolatore di tensione che rende molto più economico il sistema di alimentazione. Il generatore di clock infine è provvisto di quarzo di precisione da 1MHz.

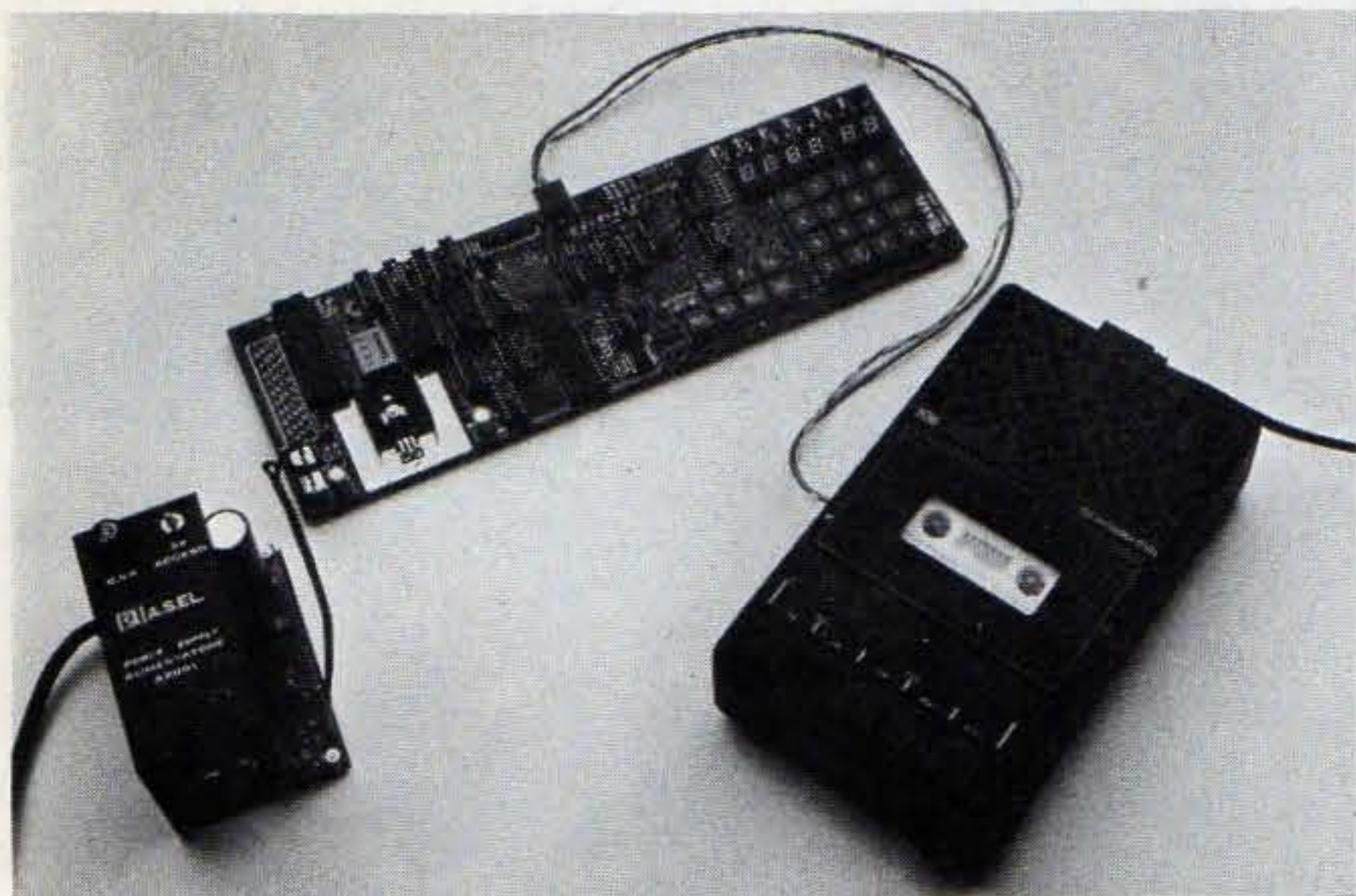



Figura 2 - AMICO 2000A collegato ad un registratore a cassette.

In linea con il formato Europa delle schede di espansione del sistema, questa unità è montata su uno stampato che ha un'altezza di 100mm (come l'altezza delle schede Europa), e una lunghezza di 300mm.

Come si può espandere il sistema

La scheda base AMICO 2000A che si è appena descritta è quindi un microcomputer completo che può soddisfare le esigenze di una gran parte di utilizzatori non interessati ad andare oltre l'apprendimento del software del 6502 e del linguaggio macchina.
La scheda può vivere da sola o può essere anche la base di un sistema di notevoli dimensioni. Vediamo di seguito le schede previste per espandere le capacità del microelaboratore.
Scheda-madre di BUS. É capace di accettare fino a 9 schede formato Europa (100x160mm) e connettori a 64 poli. La scheda è completa di circuiti di buffer e terminazioni di linea.
Alimentatore di potenza. Previsto per alimentare il sistema AMICO 2000 nella massima configurazione, esso fornisce: +5V, 6A; +15V, 1A; -15V, 1A.
Scheda di espansione RAM/EPROM. Ogni scheda può contenere fino a 4kbyte di RAM e 4kbyte di EPROM. La circuiteria è completa di buffer di dato e indirizzo e di decodifica di indirizzo.
Scheda di espansione RAM. Fino a 16kbyte con buffer e decodifica.
Scheda di interfaccia video. Contiene il monitor di gestione ed è provvista di connettore per tastiera ASCII. Ha la possibilità di generare caratteri semigrafici oltre che di presentarli in negativo.
Tastiera ASCII standard TTY. Completa di codifica con uscita parallela.
Scheda di interfaccia di comunicazione seriale. Standard RS 232/ loop di corrente e interfaccia parallela per stampante. La scheda è completa di generatore di baud rate e di due timer/counter.
Scheda di I/O digitale. È particolarmente importante, per applicazioni del microelaboratore in controlli industriali e di processo. Questa scheda è prevista in due versioni: A) optoisolata con 16 input e 16 output; B) con 32 input-output non optoisolati.
Scheda di I/O analogico. Ha 8 canali di input con risoluzione a 8 o 12 bit, multiplexer di ingresso e sample & hold con tempo di conversione di circa 30μsec; 2 canali di output a 8 bit con tensione di uscita 0÷1V e 0÷10V fondo scala.
Per racchiudere tutto il sistema in un insieme compatto viene fornito un contenitore metallico con cestello portaschede formato Europa e vano per l'alimentatore di potenza.
Questo il sistema AMICO 2000 nella massima configurazione hardware annunciata. Al momento non tutte le schede sono già disponibii per pronta consegna.
Contemporaneamente all'hardware, il costruttore curerà in particolar modo la parte didattica con la pubblicazione di articoli e libri sul sistema che saranno molto orientati all'aspetto applicativo anche per quanto riguarda il software. Sempre nell'ambito del software, saranno messe in commercio delle cassette magnetiche con programmi preregistrati. Una importante iniziativa che infine vogliamo segnalare è la prossima organizzazione di seminari didattici e di aggiornamento sull'impiego del sistema AMICO 2000 che verranno tenuti degli stessi progettisti. ■

FEEDBACK

Sistemi didattici l'MMD1

Spett. Redazione

ho letto con interesse l'articolo "Introduzione al computer" apparso su BIT n° 2 a cura della Vostra redazione, e desidero congratularmi con gli autori per la chiarezza con cui sono riusciti a raggiungere gli scopi, sintetizzati nell'ultimo paragrafo.

Salta però all'occhio, nel penultimo paragrafo dove si accenna ai microcomputer didattici, un'asserzione che riteniamo lesiva nei confronti del microcomputer MMD1.

Che l'MMD1 sia il veterano dei microcomputer didattici, siamo daccordo: esso, infatti, è stato il primo. Ma da qui a tacciarlo di vetustà ed accantonarlo in due e due quattro, mi sembra conclusione poco obiettiva.

Pertanto ritengo doveroso fare le seguenti precisazioni.

1. L'MMD1, anche se nato da qualche anno, deve ancora vivere i suoi momenti di vero successo qui in Italia, dove il mondo dei microcomputer è appena iniziato e dove quindi ci sono ancora moltissime persone che esigono un micro veramente didattico, corredato di chiari testi esplicativi, e non delle specie di sistemi di sviluppo che, casomai, si potranno raccomandare ad hobbisti iniziati, ma mai a studenti o a principianti.
2. L'MMD1 vanta un corredo di accessori e periferiche, che nessun altro microcomputer didattico offre, e che consente una logica espansione funzionale e di potenza che permette a chi insegna di portare agevolmente l'allievo dal concetto di bit alla soglia dei grossi computer. Citiamo fra i tanti: espansione di memoria fino a 64K RAM e PROM, disponibilità di BASIC in PROM, possibilità di interfacciamento con TTY o registratore audio, disponibilità di tastiera alfanumerica e di monitor, possibilità di conversione da microprocessore 8080A a microprocessore Z-80.
3. L'MMD1 è regolarmente utilizzato, come sussidio didattico, in moltissimi corsi organizzati negli USA da Università, consulenti, riviste, e in corsi "in house" presso aziende quali Eastman, ITT, Exxon, IBM. Recentemente è stato prescelto dall'autorevole Massachusetts Institute of Technology per un corso sui microcomputer offerto da questa università su videocassetta.
4. L'MMD1, infine, è corredato dei famosi BUGBOOKS V° e VI°, che in pochi mesi sono giunti ad esaurire la seconda ristampa; il che ne testimonia l'indiscusso valore. Questi libri rappresentano infatti, per l'MMD1, un bagaglio didattico di valore inestimabile e dal quale non può prescindere il neofita del micro: grazie alla loro originalità e chiarezza, questi libri stanno facendo veramente testo, e non a caso il nanocomputer NB-Z80 della SGS-ATES, da Voi citato nell'articolo, è corredato di libri chiaramente ispirati ai BUGBOOKS e ai quali gli stessi autori hanno collaborato. Questa è un'ulteriore conferma della validità dei principi che hanno ispirato la nascita dei Bugbooks e del loro naturale primogenito, l'MMD1.

Sono certo vorrete comunicare ai lettori di BIT questa mia precisazione, che ritengo necessaria nel rispetto di una rigorosa informazione su quanto propone oggi in Italia una didattica seria ed aggiornata.

Microlem divisione didattica
Dott. Giuseppe Fontana

La lettera del dott. Fontana, in sostanza, approfondisce più di quanto era nelle intenzioni degli autori dell'articolo a cui fa riferimento il discorso sui sistemi didattici, in particolare, sulle caratteristiche dell'MMD1.
Quindi nessuna volontà "lesiva" soprattutto quando si pensa all'attenzione con cui la casa editrice Jackson segue il problema dell'aggiornamento tecnico in Italia nel campo dell'elettronica professionale e della didattica e di cui fanno fede, tra le altre pubblicazioni, per l'appunto i Bugbook V e VI finalizzati al supporto dell'MMD1, nonchè i Nanobook, realizzati in collaborazione con la SGS-ATES, di prossima edizione sempre dalla Jackson e destinati a svolgere la medesima funzione relativamente al Nanocomputer NB-Z80 della SGS-ATES.
A questo punto non resta che cogliere l'occasione per sollecitare i lettori di BIT ad intervenire con consigli, pareri, critiche, contributi al discorso sul mondo dei computers, che speriamo di realizzare in maniera sempre più interessante.

GOLF

Per giocare a golf col computer occorre, al solito, correggere degli errori.
Nel listing di pag. 71-72 di BIT n° 2 devono essere apportate le seguenti modifiche

```
120     PRINT "I CATTIVI COLPI NEL GIOCO DEL GOLF COMPRENDONO:"
....................
270     IF F=1 THEN 310
....................
292     IF S1>P+2 THEN 297
293     IF S1=P THEN.299
....................
350     IF L(0)>5 THEN 1190
360     PRINT "TIRO DI"; D1; "METRI, CIOÈ"; D2; "METRI DALLA BUCA."
....................
472     GOTO 80
....................
830     Z=(RDN (1)/.8)*(2 H+16)*ABS (TAN (D1*.0035))
840     D2=INT (SQR (Q^2 + ABS (D-D1)^2))
....................
1170    D1=INT (.85*D1)
....................
1381    PRINT "SCEGLIETE IL COEFF% DI DISTANZA PER IL TIRO."
1382    PRINT "PUTTER (DA 1 A 13)."
....................
1700    DATA 361, 4, 4, 2, 389, 4, 3, 3, 206, 3, 4, 2, 500, 5, 7, 2
1702    DATA 408, 4, 2, 4, 359, 4, 6, 4, 424, 4, 4, 2, 388, 4, 4, 4
....................
1160    PRINT "IL PUNTEGGIO MEDIO TOTALE SU ";F-1;" BUCHE
        É"; G3;
```

MICROCOMPUTER SU SCHEDA SINGOLA

AIM 65

AIM 65:
il microcomputer che ha nella sua grande versatilità d'impiego il suo maggior pregio: sistema di sviluppo, controllo di processo, tester, terminale, sistema di istruzione... e poi basta solo un po' di fantasia per trovare altre mille utili applicazioni.
Anche il prezzo è quanto mai interessante!
L'AIM 65 è completo di: stampante caratteri ASCII 20 colonne – display 20 caratteri ASCII – interfaccia per due audio cassette e TTY – tastiera completa di tipo terminale – 1 K o 4 K byte RAM – bus espandibile esternamente.
Firmware: – monitor – debugger (trace, break points) – assembler – disassembler – text editor – basic.

Dott. Ing. Giuseppe De Mico s.p.a.

20121 MILANO
Via Manzoni, 31
Tel. (02) 653131 - Telex: 312035
Telegr.: Twinrapid
Uffici regionali:
Roma/Torino/Bologna/Padova

Interactive Microcomputer

R6500 HARDWARE MANUAL

AIM 65 USER'S GUIDE

AIM 65 MONITOR PROGRAM LISTING

R6500 PROGRAMMING MANUAL

Rockwell